Problem Determination -- table of contents

8: Problem determination

This section provides information about resources and techniques to help you identifyand respond to problems. You can perform problem determination at different levels within your system.Several resources are available for identifying problems:

- Logs
- Trace files
- Messages
- Tools

In order to identify a problem, it is important to understand both the topology of the system and howyour application fits into this topology. See WebSphere Structure in this section. Consider the followingquestions:

- Are all the components installed successfully?
- What is your application attempting to do?
- How is your application deployed?
- What technology is used to connect to back-end systems?
- Can you re-create the problem?
- What resources best identify the problem?

Next, choose the diagnostic tasks that can help you identifythe component within WebSphere Application Server or within your application that iscausing the problem. Diagnostic tasks include:

- Determining what tuning parameters to specify
- Identifying error messages
- Locating logs and trace files
- Determining whether system and server classpath settings are set correctly
- Identifying failing product components
- Identifying appropriate tools for a problem
- Understanding how to invoke and use available tools

WebSphere Structure topology

This sectioin describes the basic elements of WebSphere Application Server. Becoming familiarwith this information is useful preparation for using the process, resources and diagnostics forproblem determination.

8.1: Problem determination vs. tuning

This section describes a summary of the difference between problem determination and tuning.Problem determination and tuning are closely related topics, each having the same outcome: a betterperforming product. You might perceive tuning as a subset of problem determination.

Understanding the difference between problem determination and tuning is important. Knowing when to usetuning and when to use problem determination will save you time.

Problem determination

Problem determination is the process of determining the source of a problem; for example, a program component,machine failure, telecommunication facilities, user or contractor-installed programs or equipment,environmental failure such as a power loss, or user error.

Tuning

Tuning is the process of adjusting an application or a system to operate in a more efficient manner in thework environment of a particular installation.

In other words, problem determination fixes functional problems, while tuning alleviates slow processes.

The WebSphere Performance and Tuning Guidedescribes the parameters that should be modified to create an optimum product environment.

For Windows NT users, the Resource Analyzer Toolwill help you monitor and tune your system.

- 9.1: WebSphere Application Server Tuning Guide

8.2: How to view messages

There are three ways to view messages:

- Console Messages area - located on the administrative GUI
- Administrative and application server logs
- Activity log

Console messages area

The console messages area, located at the bottom of the WebSphere Administrative Console GUI,provides tracing information. The console messages area shows all messagesfor a given domain. The console messages can be disabled as to not use the facility.

The com.ibm.ws.ras.SeriousEventEnable property allows you to specify whether or not serious events (audit, warning, error, fatal, terminate) are written to the the administrative Database. If writing of serious events to the administrative database is disabled, the corresponding messages will not be available for display in the administrative console.

This property is located in the <install_dir>/properties/logging.propertiesfile. A process must be restarted for the change to take effect.

See the InfoCenter article 6.6.0.1: Using the Java administrative console to learn more information about the messages area.

Event viewer

The event viewer maintains a collection of the most recent message events. You can configure the event viewer to show any combination of audit, fatal, terminate, and warning message events. See the InfoCenter article 6.6.0.1: Using the Java administrative console to learn more information about the event viewer.

The property com.ibm.ws.ras.UnitOfWork allows you to specify whether or not a correlation ID should be generated and included in message events and diagnostic trace entries. If set to true, each application client request is assigned a unique identifier that is propagated to all servers touched as part of servicing that request. Thisallows correlation of events across multiple server processes.

This property is located in the <install_dir>/properties/logging.propertiesfile. A process must be restarted for the change to take effect.

Administrative and application server logs

These logs will show the messages for a given server process. Use your favorite text editorto view these logs. See the Logs section for more information.

Activity log

The activity log will show messages for a given server process. Use the Log Analyzer or the showlogfacility to view the messages. See the Logs section for more information.

8.2.1: How to View Messages

You can view messages in the console messages area and with theevents viewer.

Console Messages area

The Console Messages area, located at the bottom of the WebSphere Administrative Console,provides the highest level of tracing information. When a message eventoccurs, it is displayed in the Messages area.

When you require more information than the Messages area provides, review the messagesrecorded in the Event Viewer.

Event Viewer

The Event Viewer maintains a collection of the most recentmessage events. Youcan configure the Event Viewer to show any combination of audit, fatal,terminate, and warning message events.

The Event Viewer "Log Limit" property specifieshow many event records to keep. They are stored in the administrativerepository (database). If your database is becoming too full, setthe Log Limit to the minimum value that is reasonable for your environment.

Access the Event Viewer from the Console menuon the menu bar of the administrative console.

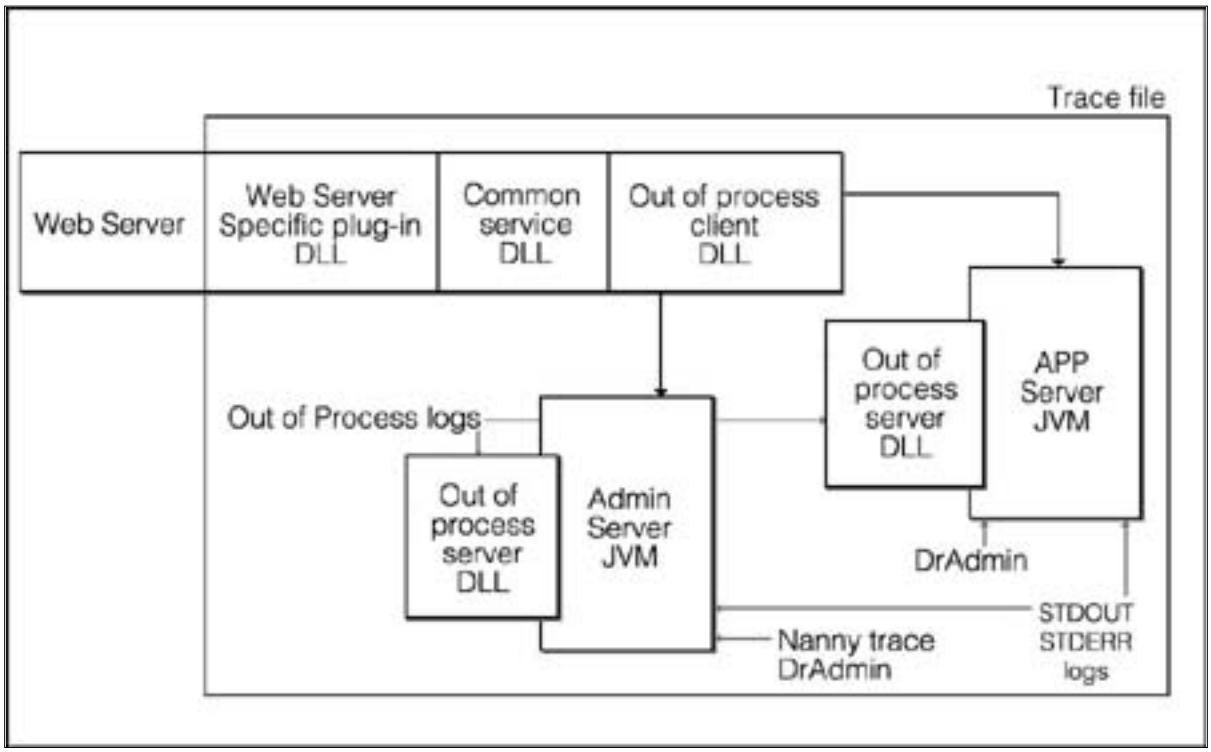
8.3: Logs

WebSphere Application Server provides many error logs to help you diagnoserun-time problems. This section describes these error logs telling you where to find andhow to format the files. The logs are:

- activity.log
- stderr.log
- stdout.log
- oop logs for the application server
- oop logs for the admininstrative server
- NT wasdb2.log
- NT wssetup.log
- UNIX WebSphere.instl

The tools required to process some of these logs (as well as some of the trace logs) are described inUsing Internal Tools. You can also refer to Problem determination hints and tips for additional tips on the use and processing of some of these error logs. If you need to report aproblem to IBM, you might need to gather some of these error logs and send them to IBM for diagnosis; for moreinformation, refer to How to report a problem to IBM.

View the following graphic for a description of the log and trace points inWebSphere Application Server:



InfoCenter Home >
8: Problem determination >
8.3: Logs **<- you are here**

Activity log for problem determination

The activity log captures events that show a history of WebSphere Application Server'sactivities. Some of the entries in the log are informational, while others report on systemexceptions, such as returned CORBA exceptions.

When you encounter WebSphere Application Server run-time errors, you will often find it usefulto use Log Analyzer to read the activity log and try to diagnose the problem yourself.When you need assistance from IBM to help you diagnose problems, you will be asked to providethe formatted activity log output to IBM.

Location of the activity log

There is one activity log for each host machine. The activity.log file resides in the logsdirectory of where the product is installed. All application servers, including the administrativeserver, write error records to this file. The activity.log file is a binary file and cannotbe viewed with an ascii editor. You can view the activity.log file in one of two ways:

- Log Analyzer
- showlog

How to view the activity.log file with Log Analyzer

1. Change the directory to:
   product_installation_root/bin
2. Run the waslogbr script file, which is called:
   - waslogbr.bat on Windows NT
   - waslogbr.sh on Unix systems

   It needs to be run from the bin directory cited above.
   This will start the Log Analyzer graphical interface.
3. In the interface:
   1. Select File>Open.
   2. Navigate to the directory containing the activity.log file.
   3. Select the activity.log file.
   4. Select Open.

How to view the acitivity.log file on a remote machine using showlog

The Log Analyzer cannot be used to view remote files.An alternate tool named showlog can be used instead of Log Analyzer to format the activity.log file for viewingwhen no GUI display capabilities are available.

showlog.bat or showlog.sh is a script/batch file that can be found in the bin directoryof the WebSphere Application Server installation. Follow these instructions to use showlog:

1. Change directory to:
   product_installation_root/bin
2. Run the showlog tool with no parameters to display the usage instructions:

- On Windows NT, run showlog.bat
- On Unix systems, run showlog.sh

Examples:

- To direct the activity log contents to stdout, use the invocation:
  showlog activity.log
- To dump the activity.log to a text file that can be viewed using a text editor, use the invocation:
  showlog activity.log textFileName

Changing activity.log file size

In the course of using Log Analyzer, you might have to set the maximum activity.log file size.The activity.log file grows to a predetermined size and then wraps. The default size is 1MB. Followthese steps to change the log size:

1. Open the properties file in a text editor:
   product_installation_root/properties/logging.properties
2. For the SHARED_LOG_LENGTH property, specify the value you would like in Kilobytes (KB). If anindividual size is entered, the default size is used.
   - Example: To change the log size to 2MB, enter in the line:
     SHARED_LOG_LENGTH=2048 (do not use spaces)

The size change will take effect at the next server startup.

See other related tasks, including changing the port of the logging service, diagnosing port conflictsand confirming the creation of the activity.log file in the Log Analyzer section.

stderr and stdout logs for problem determination

The stderr and stdout logs capture events presented through two of the three standard I/O streams, or:

- stdin - arguments entered with a command or program
- stdout - output displayed to the user
- stderr - errors thrown by the code

In WebSphere Application Server, the stdout and stderr logs are created for:

- Application servers
- Servlet redirectors

The application server stderr and stdout logs contain application server communication.Output from `System.err.println` and `System.out.println` statements in theservlet code also appear in the application server stdout and stderr logs.

InfoCenter Home >
8: Problem determination >
8.3: Logs <- **you are here**

Out-of-process(oop) logs for problem determination

There are two types of out-of-process logs: those created for administrative servers and those created forapplication servers.These logs contain error and informational messages generated from the native code portion of the out-of-processengine. This information reflects server startup and server status change requests (start/stop/restart).

The default log file mask setting for the oop logs is **error**.If these logs have a file length of zero, no error messages were generated during the server status change requests.

To change the default log file file mask setting:

1. Select the process to trace, for example **Servlet Engine**.
2. Click the Advanced tab for the Servlet Engine.
3. Click the Settings button.
4. Choose the log file mask from one of four options.
5. Click the Restart Plug-in button

The oop log files are now dated so the file format is consistent with that of the HTTP Server trace logs. Forexample, an administrative server oop log file name is:`adminserver_native.log.was.oop.Thu_May_26_23.26.20.2000`

NT wasdb2 log

This log is created when the DB2 database, *was*, is configured. Review this log to determine if the*was* database was created correctly and if WebSphere Application Server can connect to it.Errors creating the system management repository tables will be logged in this file.

NT wssetup log

This log is created during the install process. Review this log to ensurethe install process was successful. The install process consists of:

- Verifying prerequisites
- Downloading files
- Updating the configuration files for both WebSphere Application Server and the Web server

UNIX WebSphere.instl

On AIX and Solaris, a native install of WebSphere Application Server generates theWebSphere.instl log that is located in the `/tmp` directory.

Information on the WebSphere Application Server install process on HP isplaced in the HP system log, swagent.log, that is located in the`/opt/WebSphere/AppServer/var/adm/sw` directory.

See the file, Sample Logs, for examples of WebSphere.instl andswagent.log logs.

See the file, WebSphere Application Server Directories, for a listing of filesand directories in Version 3.5. This listing may help with problem determination.

The Related information links to instructions and additional detail aboutenabling and interpreting logs.

9.3.2: Sample Logs

[illegible]

stderr and stdout logs

[illegible]

oop logs

[illegible]

8.4: Traces

Traces are just logs.Traces and logs differ in that you must turn traces on to see output in atrace file. Logs are always enabled and log entries are automatically generated.

Tracing occurs at two levels:

- On administrative servers (usually one per node)
- On application servers

You can enable trace for each Java™ class in the system.

The trace subsystem does not trace user code (such asservlets or EJB components) unless `System.err.println` or `System.out.println`statements are added to the code. Output from the `println` statements appears eitherin the application server stdout or stderr logs.

See the stdout and stderr logs description for more informationon stdout and stderr logs.

Beginning with WebSphere Application Server Version 3.0,an object level debugger is provided with the product to trace and debug user code.See the Object Level Tracing and Debugging (OLT and OLD) section for information on objectlevel tracing.

Trace and log entry format

Since trace is just another log, both a WebSphere Application Server log entry anda trace entry will have the same format. The following example of a log entry illustratesthe format:

Log entry example: **[00.07.11 22:47:12:191 EDT] 53ccc3c5 ActiveEJBCont W Could not create beantable xxx**

The following table includes a description of each of part of the log entry:

[00.07.11 22:47:12:191 EDT]	53ccc3c5	ActiveEJBCont	W	Could not create bean table	xxx
TS: The timestamp in fully qualified date (YYMMDD), Time (Millisecond precision), and Time zone format.	**TID**: The thread ID or the hash code of the thread issuing this message.	**COMPONENT**: The short name of component issuing this message.	**LEVEL**: The level of the message or trace. Possible levels are: • > Entry to a method (debug) • < Exit a method (debug) • **A** Audit • **W** Warning • **X** Error • **E** Event (debug) • **D** Debug (debug)	**MESSAGE**: The text of the message.	**ARGUMENTS**: Optional message arguments.

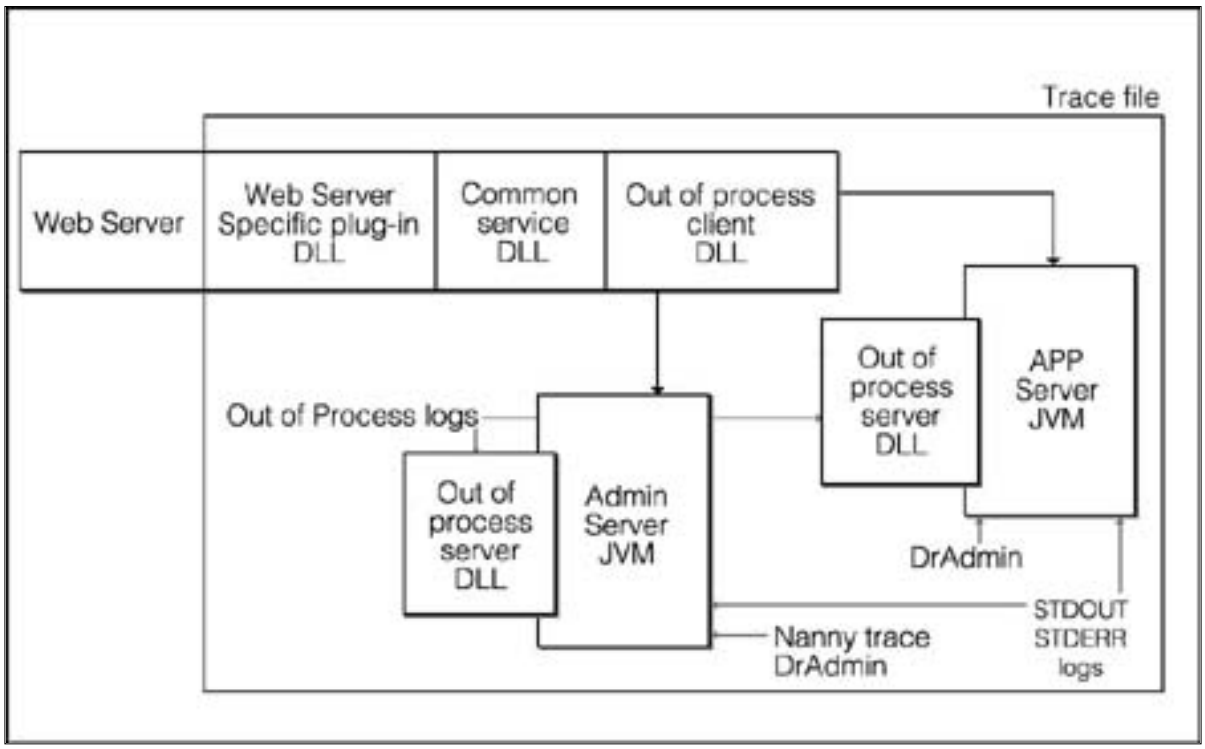
			• **T**Terminate (exits process) • **F** Fatal (exits process)		

Types of traces

The following are the traces you will find in WebSphere Application Server:

- Trace file
- UNIX Nanny trace
- DrAdmin

View the following graphic for a description of the log and trace points inWebSphere Application Server:



Trace file

The trace file provides trace entries on the interaction of various WebSphere ApplicationServer components with the administrative server.Use the trace file to identify a problem and to review events preceding the error situation.

Note: Always review trace entries prior to the error.Trace entries recorded after the error has occurred represent program recovery and will nothelp with problem determination.

Review the Collecting traces section for additional tracedocumentation.

Trace file samples

UNIX Nanny trace

On UNIX platforms, the nanny process starts the administrative server.The nanny.maxtries parameter in the admin.config file tells the nanny process how many timesit should attempt to restart the administrative server.

On Windows NT, the nannyservice is part of the IBM WebSphere Administrative Server service that is registered with theoperating system.Starting the IBM WebSphere Administrative Server service invokes adminservice.exe. If theservice does notstart, verify that:

- The service was installed and is available from `Start > Settings > Control Panel > Services`
- The userID under which WebSphere Application Server was installed has service privileges

If the nanny process fails to start the administrative server on UNIX or if theIBM WebSphere Administrative Server service does not start on Windows NT, you can bypass the nannyfunction and just start the administrative server. Follow these steps to start the administrative server:

1. Go to the <WAS_root>bin/debug directory
2. Invoke adminserver.sh on UNIX or adminserver.bat on Windows NT

   Note: Starting the administrative server without using thenanny function means that nothing is monitoring the administrative server. If it fails in thisstate, nothing will restart it.

A nanny trace is only available on UNIX platforms.

On Windows NT, use the Event Viewer to view entries related to the WebSphere nanny service. Followthese steps to view the Event Viewer :

1. Select Start > Programs > Administrative Tools
2. Select Event Viewer
3. View events related to WebSphere Application Server

Nancy trace samples

DrAdmin trace function

The DrAdmin function generates thread dumps.

On UNIX platforms, the IBM JDK allows users to send signals to force javacore.txt filesto be created in the application server's working directory.The application server continues to run and a sequence of javacore files are created. These filescan help in debugging "loop" or "system hang" problems.

To generate thread dumps similar to the javacore files, especially on a Windows NTplatform, use the DrAdmin function.

A unique DrAdmin port is generated each time an application server starts. To generate a threaddump for that port:

1. View the console messages area or the trace filefor message SMTL0018I "DrAdmin available on port."
2. Enter the following command:
   `DrAdmin <port number> -dumpThreads`
3. Review the stderr log for the thread dump.

After installing and starting WebSphere Application Server, you will see DrAdmin entries in theconsole messages area. These entries appear regardless of the options specified during installation,and have the

following format:
DrAdmin available on port 1,055

DrAdmin entries are also recorded in the trace file.To locate the DrAdmin entry in the trace file:

1. In the <WAS_ROOT>**logs** directory, open the trace file.
2. Go to the bottom of the trace file and then scroll up until you locate the following entry:`DrAdmin available on port xxxx`

What is DrAdmin?

DrAdmin is a service, provided by each of the servers, to enable and disable tracing. Each time a server starts, DrAdminregisters itself on a different (next available) port number. There are no output messages associated with DrAdmin. TheDrAdmin entries in the console messages area are generated to tell users the port number where DrAdmin is listening.

When to use DrAdmin?

You should always use the administrative console trace facilities to debug a problem.DrAdmin is a diagnostic tool to be used when you have no other choices. It was designed to provide IBM support personnelwith another way of obtaining traces. Use DrAdmin in these situations only:

- When input to the administrative console is not accepted
- When the administrative server is in a wait state
- When the administrative server is not responding (e.g., in an infinite loop orhung state)
- When you have to dump the thread stacks in a server
- When the administrative client topology tree disappears

Note: DrAdmin is an internal interface that is used to assist users with problem determination.As an internal interface, it is subject to change at any time, and there is no nationallanguage support for it.

How to use DrAdmin?

The DrAdmin interface is the same on all platforms. Since DrAdmin is another way ofturning on a trace, the tracing mechanism is the same as the one used by theadministrative console trace facilities.Therefore, whether you are looking at the trace file or a DrAdmin output file,the trace entries will have the same format.

See the DrAdmin samples to learn how to invoke DrAdmin.

To specify the DrAdmin trace output in the admin.config file:

- In the admin.config file located in the <WAS_ROOT>bin directory,append the following property where the file name consists of an absolute path and the file that you want to create:`com.ibm.ejs.sm.adminServer.traceOutput=<filename>`
- Restart the administrative server.The trace file specified in the admin.config file should be created during startup of theadministrative server. The administrative server might not start if the path specified forfile name does not exist, or if the directory permissions where the trace fileis to be created, are incorrect or insufficient.
- If the administrative server starts but the trace file is not created, ensure the traceOutputparameter in the admin.config file is correct, and the path to the file exists.

DrAdmin samples

For more information on traces, see file What are messages, logs and traces?

8.4.1: Trace samples

Select one of the following traces to view sample output:

- Trace file
- UNIX Nanny trace (available on UNIX platforms only)
- DrAdmin

Trace file

The following trace fragment is an example of a trace file.Use this trace to debug server startup or shutdown problems:

```
[00.07.17 15:59:57:200 EDT] f0c45c4c AdminServer   A Initializing WebSphere Administration server[00.07.17 15:59:57:230 EDT]
a9d5dc4e DrAdminServer A DrAdmin available on port 1,038[00.07.17 16:00:22:457 EDT] f0c45c4c SASConfig     A SAS
Property:com.ibm.CORBA.principalName has been updated[00.07.17 16:00:25:191 EDT] f0c45c4c InitialSetupI A Creating Sample Server
Configuration[00.07.17 16:00:29:797 EDT] f0c45c4c JDBCDriverCon A Importing JDBCDriver : Admin DB Driver[00.07.17 16:00:31:209
EDT] f0c45c4c JDBCDriverCon A Installing JDBC Driver: Admin DB Driver on node db[00.07.17 16:00:31:530 EDT] f0c45c4c
JDBCDriverCon X Failed to install JDBC Driver Admin DB Driver  on node db.OpException
com.ibm.ejs.sm.exception.JDBCDriverAlreadyInstalledException[00.07.17 16:00:31:770 EDT] f0c45c4c DataSourceCon A Importing
DataSource : Default DataSource[00.07.17 16:00:32:451 EDT] f0c45c4c NodeConfig    A Importing Node : db[00.07.17 16:00:32:962
EDT] f0c45c4c ApplicationSe A Importing ApplicationServer : Default Server[00.07.17 16:00:33:823 EDT] f0c45c4c ContainerConf A
Importing Container : Default Container[00.07.17 16:00:35:746 EDT] f0c45c4c EJBConfig     A Importing EJB : HitCount
Bean[00.07.17 16:00:37:859 EDT] f0c45c4c EJBConfig     A Importing EJB : BeenThere Bean[00.07.17 16:00:39:271 EDT] f0c45c4c
ServletEngine A Importing ServletEngine : Default Servlet Engine[00.07.17 16:00:40:843 EDT] f0c45c4c WebApplicatio A Importing
WebApplication : default_app[00.07.17 16:00:44:088 EDT] f0c45c4c ServletConfig A Importing Servlet : snoop[00.07.17 16:00:44:248
EDT] f0c45c4c ServletConfig W Updating Servlet : snoop, since it was already created[00.07.17 16:00:48:604 EDT] f0c45c4c
ServletConfig A Importing Servlet : hello[00.07.17 16:00:48:694 EDT] f0c45c4c ServletConfig W Updating Servlet : hello, since it
was already created[00.07.17 16:00:51:508 EDT] f0c45c4c ServletConfig A Importing Servlet : ErrorReporter[00.07.17 16:00:51:609
EDT] f0c45c4c ServletConfig W Updating Servlet : ErrorReporter, since it was already created[00.07.17 16:00:53:982 EDT] f0c45c4c
ServletConfig A Importing Servlet : invoker[00.07.17 16:00:54:182 EDT] f0c45c4c ServletConfig W Updating Servlet : invoker, since
it was already created[00.07.17 16:00:56:586 EDT] f0c45c4c ServletConfig A Importing Servlet : jsp10[00.07.17 16:00:56:806 EDT]
f0c45c4c ServletConfig W Updating Servlet : jsp10, since it was already created[00.07.17 16:01:02:825 EDT] f0c45c4c WebApplicatio
A Importing WebApplication : admin[00.07.17 16:01:05:428 EDT] f0c45c4c ServletConfig A Importing Servlet : install[00.07.17
16:01:05:539 EDT] f0c45c4c ServletConfig W Updating Servlet : install, since it was already created[00.07.17 16:01:07:982 EDT]
f0c45c4c ServletConfig A Importing Servlet : jsp10[00.07.17 16:01:08:092 EDT] f0c45c4c ServletConfig W Updating Servlet : jsp10,
since it was already created[00.07.17 16:01:14:271 EDT] f0c45c4c ServletConfig A Importing Servlet : file[00.07.17 16:01:14:361
EDT] f0c45c4c ServletConfig W Updating Servlet : file, since it was already created[00.07.17 16:01:16:865 EDT] f0c45c4c
ServletConfig A Importing Servlet : invoker[00.07.17 16:01:16:975 EDT] f0c45c4c ServletConfig W Updating Servlet : invoker, since
it was already created[00.07.17 16:01:19:439 EDT] f0c45c4c ServletConfig A Importing Servlet : ErrorReporter[00.07.17
16:01:19:529 EDT] f0c45c4c ServletConfig W Updating Servlet : ErrorReporter, since it was already created
```

UNIX Nanny trace

The following trace fragment is an example of a nanny trace.Use the nanny trace to monitor administrative server events:

```
[00.07.17 17:05:00:032 EDT] 1fa4cc16 Nanny         > main"admin.config"[00.07.17 17:05:00:032 EDT] 1fa4cc16 Nanny         >
Initial admin server startup..[00.07.17 17:05:06:231 EDT] 1fa4cc16 Nanny         < Initial adminserver startup
successful..[00.07.17 17:05:06:321 EDT] 1fa50b45 Nanny         > run : AdminServerMonitorThread[00.07.17 17:05:06:321 EDT]
1fa50b45 Nanny         E AdminServerMonitorThread: Waiting for process 1719 to terminate.
```

DrAdmin

NT To invoke DrAdmin:

1. Go to the <WebSphere\AppServer\bin\debug> directory.
2. Copy adminserver.bat to DrAdmin.bat

   UNIX **Note:**On Unix platforms,the adminserver.bat file is adminserver.sh.Copy adminserver.sh to DrAdmin.sh.
3. Replace the following line in the DrAdmin.bat file:
   %JAVA_HOME%\bin\java -mx128m com.ibm.ejs.sm.server.AdminServer -bootFile %WAS_HOME%\bin\admin.config %restart% %1 %2 %3 %4

   with

   %JAVA_HOME%\bin\java com.ibm.ejs.sm.util.debug.DrAdmin %1 %2 %3 %4 %5 %6 %7 %8 %9

4. Save and close the DrAdmin.bat file
5. From a command prompt in the <WebSphere\AppServer\bin\debug> directory, type DrAdmin [options]where options are:
    - -help [shows the help message]
    - -serverHost <Server host name> [Specify the host name of the server... defaults to local host]
    - -serverPort <Server port number> [Required... enter the port number where DrAdmin is listening]
    - -setTrace <Trace specification> [Specify any valid traceString, for example,"com.ibm.ejs.sm.*=all=enabled"]
    - -setRingBufferSize <Number of ring buffer entries in k> [Specify the number of trace entries to store in themain memory buffer... the default is 8k]
    - -dumpRingBuffer <Name of file to dump the ring buffer> [Defaults to file name JMONDump.xxxxxxxxxxxx where xxxxxxxxxxxx isa combination time of day and unique PID identifier extension]

      NT **Note:** OnWindows NT, if the administrative server is started as a service, the default DrAdmin dump file will be located in the<Winnt\system32> directory.
    - -dumpState <dumpString> [Specify a unique identifier for this dump]
    - -stopServer [Stops the administrative server]
    - -stopNode [Does not apply unless the node is connected to the administrative server]
    - -dumpThreads [Dumps the threads in the server]

Another example of implementing a DrAdmin trace:

NT On Windows NT:

1. Create a DrAdminRun.bat file that contains the following information:`set CLASSPATH=C:/jdk1.1.7/lib/classes.zip;C:/WebSphere/AppServer/lib/ujc.jar;C:/WebSphere/AppServer/lib/ejs.jar;C:/WebSphere/AppServer/lib/admin.jarecho %CLASSPATH%echoechojava -classpath %CLASSPATH% com.ibm.ejs.sm.util.debug.DrAdmin -serverPort %1 -setTrace %2=%3=%4`
2. Invoke DrAdminRun.bat with the port number and the trace string. Use the port number from DrAdmin entryin the trace file, xxxx. Your input from a command prompt will be:`DrAdminRun xxxx com.ibm.ejs.* all enabled`
3. Start administrative client with the debug option by invoking adminclient.bat from the WAS_ROOT bin directory:`adminclient debug`

UNIX On UNIX platforms:

1. Create a shell script file DrAdminRunthat contains the following information:`# modify classpath as appropriate for platform/environment# run as follows: sh DrAdminRun <server port> <trace spec>export CLASSPATH=/usr/jdk_base/lib/classes.zip:/usr/WebSphere/AppServer/lib/ujc.jar:/usr/WebSphere/AppServer/lib/ejs.jar:/usr/WebSphere/AppServer/lib/admin.jarecho $CLASSPATHechoechojava -classpath $CLASSPATH com.ibm.ejs.sm.util.debug.DrAdmin -serverPort $1 -setTrace $2`**Note:** Verify the CLASSPATH is correct for your environment.The script example was written for AIX. You must change the CLASSPATH for Solaris.
2. Invoke DrAdminRun with the port number and the trace string. Use the port number from DrAdmin entryin the trace file, xxxx. Your input from a command prompt will be:`sh DrAdminRun xxxx com.ibm.ejs.*=all=enabled`
3. Start the administrative client with the debug option by invoking adminclient.sh from the WAS_ROOT bin directory:`adminclient.sh debug`

8.4.1.1: Plugin Problems

HTTP Servers are legacy Web products, created at a time when theywere the only conduit between Browsers and HTML or CGI files. With the evolution ofWeb technology, users now require servers to handle servlets, JSP files and EJBs.WebSphere Application Server supports this technology but toprovide these functions, it mustintercept requests sent to the HTTP Server.

The *plugin* component extends the function of the HTTP Server.It intercepts requests and passes them toeither WebSphere Application Server or the HTTP Server. The following threefiles in the **<WAS_root>temp** directory allow the plugin to determine therequest's destination:

- <WAS_root>/temp/**rules.props**
  (Provides a snapshot of the existing topology and lists available Web resources and pathsto handle service requests.)
- <WAS_root>/temp/**vhosts.props**
  (Provides virtual hosting information that is transferred to the WebSphere ApplicationServer runtime environment.)
- <WAS_root>/temp/**queues.props**
  (Provides names of links to different servlet engines. The number of links listed inthis file vary according to the number of AppServers, clones, and other servlet engine resourcesthat are defined in the product.)

This is the high level view of the plugin process flow:

```
Browser -->  WebSphere plugin -->  HTTP Server or WebSphere Application Server
```

Typical plugin problems

Generally plugin failures are caused by missing files or an incorrectly configured HTTP Server.

To diagnose plugin problems, verify the data in the files is consistent with both the HTTP request andthe active configuration in the Servlet Engine. The plugin configuration files aregenerated periodically so a delay can occur between the time a change is made in the system andthe time the change is reflected in the configuration files.

The following error descriptions are symptomatic of plugin problems:

1. ***Servlet requests are not fulfilled***. Verify the following to determinethe cause of the problem:
   - Web server can serve HTML pages.
   - Admin console can connect to Web server.
   - Default Server is started.
   - Ensure the Web server hostname and port number are identical to the ones definedin the virtual host's alias table.
   - Ensure the appropriate **.DLL** file for the Web server is present in directory:
     <WebSphere-root>bin
2. ***Pipe broken messages appear***. Verify the following to determine the cause ofthe problem:
   - TCP/IP connection exists between Web server and WebSphere Application Server.
   - No process or thread failures occurred.
   - No access violations occurred.

How to debug plugin problems

Check for errors in the following logs, and in the ***tracefile***:

- trace.log.<http server>.<date>

- <AppServer>_stderr.log
- <AppServer>_stdout.log
- <AppServer>_native.log.<date>

See files 8.3: Logs and 8.4: Identifying the Problemfor trace and log information.

If there are no entries in the logs or tracefile. comment out the WebSphere ApplicationServer plugin in the ***httpd.conf*** file.This will help you determine if thefailure originates with WebSphere Application Server or the HTTP Server.

The WebSphere plugin property in ***httpd.conf*** is:

```
Load Module ibm_app_server_module
```

Restart the HTTP Server. If the Web Server initializes and runs, then WebSphere Application Serverhas a problem.

8.4.1.2: Servlet Redirector Problems

Servlet redirectors are used to separate the HTTP server from the WebSphere Application Server.There are different types of redirectors:

- Thick - servlet redirector resides on the same machine as WebSphere Administrative Server.
- Thin - servlet redirector runs on a separate machine from the WebSphere Administrative Server.
  A *thin* servlet redirector is useful in network configurations where the HTTP server is outside a firewall but WebSphere Application Server isbehind a firewall, or where WebSphere Application Server is located in the DMZ, a machine located between twofirewalls.

  Note: If Web Server, App server and the Server handling servlet requestsall reside on the same machine, use OSE - Open Servlet Engine, instead of a servlet redirector

Key features

The key features of servlet redirectors are:

1. Use IIOP (Internet Inter-Orb Protocol) for communication
2. Are initialized by copying the *queues.properties*, *rules.properties*,and *vhosts.properties* files from the WebSphere Application Server machine to the servlet redirector machine.
3. Use the following ports to transfer the properties files:
   - port 900 - bootstrap port
   - port 9000 - name services port
   - redirector listener port
4. Require the receiving RemoteSRP bean to be running on the WebSphere Application Server
5. When servlet redirectors are running, beside the ports previously listed, they alsorequire the following, additional ports:
   - Application Server listener port
   - Admin server nanny listener port
   - *Thick* servlet redirectors also require repository database connection ports

Typical Problems

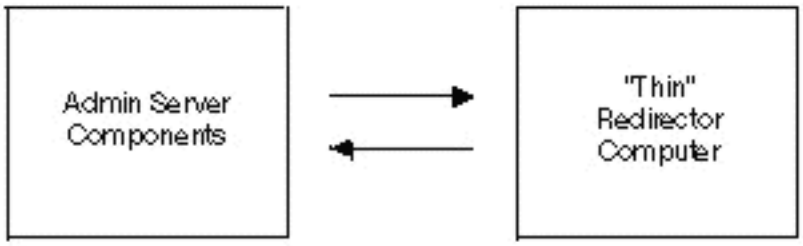
Typical problems with servlet redirectors are:

1. **Error 404** - *URL not found* appears on browser when accessing a servlet.
   The trace log for the HHTP server contains entry:
   Failure while locating a vhost for <server>
   This error may occur if:
   - The short (myserver) and fully qualified (myserver.mydomain.com) hostnames of theHTTP server are not in the **virtual host alias list.**
   - A port other than the standard port 80 is used and that port is not in the**virtual host alias list.**
2. **CORBA.COMM_FAILURE** appears when running the *thin* servlet redirector. Thiserror may occur if:
   - Syntax error exists in batch file
   - Hostname/port number is not in DNS or *hosts.properties* file on servlet redirectormachine
   - Hostname/port number is not in DNS or *hosts.properties* file on WebSphere ApplicationServer machine

- -ListenerPort parameter value is not unique
- Syntax error in -ListenerPort parameter

3. ***Unable to initialize threads: (null)*** error when running the batch/script fileto configure the plugin files.This error may occur if:
   - Some JDK other than the IBM supported JDK is in the CLASSPATH ahead of the supported JDK. To resolvethis problem, install the supported IBM JDK, and change the PATH and CLASSPATH to point to it first.
   - PATH does not point to *java.exe*.
4. When starting servlet redirector with batch/script file, get error message ***Errorlocating Remote SRP Home - Attribute Not Set***. This error occurs because:
   - RemoteSRP bean was **not** added and started on WebSphere Application Server
5. Do **not** use servlet redirectors if you must:
   - enable security on your *thin* servlet redirector machine
   - provide the real Web browser on your *thin* servlet redirector machine
   - use the *Network Address Translation* feature of firewalls so thatthe internal address of WebSphere Application Server is not available outside the firewall

   Reminder: Generally errors occur configuring the *thin* servlet redirectorfunction because required configuration files are missing. A *thin*servlet redirector machine requires the following shell script files (on UNIX platforms), or bat fileson Windows NT as well as the iiopredirector.xml file to enable the servlet redirector function.



Unix and **NT**:
iiopredirector.xml
in directory:
<AppServer root> properties

NT:
redirectorConfig.bat
redirectorStart.bat
in directory:
<AppServer root>\bin\debug

Unix:
redirectorConfig.sh
redirectorStart.sh
in directory:
<AppServer root>/bin/debug

Unix and **NT**:
iiopredirector.xml
in directory:
<AppServer root> properties

8.4.1.3: Workload Management/Cloning/Remote Administration Problems

Unlike WebSphere Application Server Advanced Edition, the Standard Edition does not supportthe Workload Management function. WebSphere Application Server Standard Edition is limited toa single physical server. However, Standard Edition provides multiple JVMs that can be mappedto multiple virtual hosts on a single HTTP Server. Therefore multiple Web sites can still be hosted usingone Standard Edition Application Server.

Description of the Workload Management function and cloning

Like the Standard Edition, WebSphere Application Server Advanced Editionprovides multiple JVMs, but it also supports distributed system management for networksand clusters of Advanced Edition Application Servers. Distributed system management is anoutcome of the the Workload Management function. This function enablescloning, or creating instances of a model. Clones can be created on a single machine or across multiplemachines in a cluster. Modifying a model automatically propagates the changes to all of its clones.By administering a model, you can simultaneously administer several copies of a server (the clones).

Scalability in a workload management environment occurs in two forms:

- ***Horizontal scaling*** - the number of concurrent connected users.
- ***Vertical scaling*** - the number of concurrent requests.

With Workload Management, requests can be routed to any server on any node with thesame results. Workload Management provides for ***Failover*** where requests are routed to othernodes (servers) when one node fails. Failover occurs without affecting throughput or reliability.

Workload Management and Cloning problem descriptions

The following resources can be cloned. Select a specific type of resource to view typicalproblem scenarios.

- Application Servers
- EJB Containers
- Enterprise Beans
- Servlet Engines
- Web Applications
- Servlets

Application Server issues

There may be some performance impacts using cloning:

1. Make small, incremental changes to your model to minimize performance hitson the client.
2. Change your model when few clients and servers are running.
3. You do not need to reconfigure server groups to reflect an unavailable server unlessthat server will be unavailable for an extended period of time. In that case, reconfigure themodel to optimize performance.

EJB Containers and Enterprise Beans

Homes of session beans have special information to indicate that these "homes" areworkload management

(WLM) enabled; that is, deployed on one or more servers. When a client accessesthe home of a session bean, the client is referred to the homeOfHomes on the Admin Server. The clientthen creates proxies for each clone using a specific workload management policy as for example,round robin.

Problems occur when:

- Client stubs are generated that are not WLM enabled. All stubs generated by VisualAge for Java areWLM enabled. For those stubs not created through VisualAge for Java, you must use the wlmjar file.
- Client stubs are WLM enabled but should not be. To create non-WLM stubs, use "rmic -iiop" and jarthe output files.
- You WLM enable a stateful session bean instead of its container.
- If all your requests are sent to the same server, verify that your client jar fileis WLM enabled. Use `jar -tvf` on the jar file, and verify the jar file containsboth x_Stub and x_BaseStub, not just x_Stub.

Servlet Engines, Web applications and Servlets

If requests are going to one server, verify that your model has more than one clone. Alsoverify that each clone has xBean deployed. This particularly applies to a *Thin Servlet Redirector*configuration since the ThinRedirector is an EJB client to the RemoteSRP bean.Also verify that the first server in the model is running. The first server must be runningfor the lookup phase to work correctly.

Remote Administration

You can remotely administer WebSphere Application Server using:

- remote admin console
- X Windows clients on UNIX machines
- Web based WebSphere admin console

Remote admin console

Use can run admin console remotely using the ***adminclient.bat*** file on Windows NT, or***adminclient.sh*** file on UNIX. See articleAdministrative models for more information on implementing a remoteadmin client.

Typical remote admin console problems are:

- ***com.ibm.ejs.util.cache.FaultException*** error occurs in the stack trace becausethe JDK running on the client machine cannot communicate with the JDK running on the Admin Servermachine. The resolution is to upgrade the backlevel JDK.
- The NAT (Network Address Translation) function in firewalls cannot be used with a remote admin client.The internal address of the Admin Server is not recognized by the admin client outside thefirewall. ***No*** circumvention exists for this problem.

X Windows clients on UNIX machines

X windows client software can run on any platform but requires a UNIX X Windows Server.(Currently the X Windows Server is only available on AIX and Solaris platforms.)

Typical X window client problems are:

- You cannot run an admin console remotely through the X windows client using anauthorized, ***non-root*** account with global security enabled.Error message, *FATAL Could not bind to the Admin Server on {0}{1}*, appears onthe screen when the adminclient .sh or .bat file is executed.***No*** circumvention exists

for this problem.

Web based WebSphere admin console

See article Web administrative console overview for informationon configuring and implementing a Web based admin console.

8.5: Identifying the problem

Please select a related topic.

8.5.1: Plug-in problems

HTTP servers are legacy Web products, created at a time when theywere the only conduit between browsers and HTML or CGI files. With the evolution ofWeb technology, users now require servers to handle servlets, JSP files and EJBs.WebSphere Application Server supports this technology, but toprovide these functions it mustintercept requests sent to the HTTP Server.

The plug-in component extends the function of the HTTP Server by interceptingrequests and passing them to either WebSphere Application Server or the HTTP Server. The followingthree files in the `<WAS_root>temp` directory allow the plug-in to determine therequest's destination:

- `<WAS_root>/temp/`**`rules.props`**
  - Provides a snapshot of the existing topology and lists available Web resources and pathsto handle service requests
- `<WAS_root>/temp/`**`vhosts.props`**
  - Provides virtual hosting information that is transferred to the WebSphere ApplicationServer runtime environment
- `<WAS_root>/temp/`**`queues.props`**
  - Provides names of links to different servlet engines. The number of links listed inthis file vary according to the number of application servers, clones and other servletengine resources that are defined in the product.

This is the high level view of the plug-in process flow:

```
Browser -->  WebSphere plugin -->  HTTP Server or WebSphere Application Server
```

Typical plug-in problems

Generally, plug-in failures are caused by missing files or an incorrectly configured HTTP Server.

To diagnose plug-in problems, verify the data in the files are consistent with both the HTTPrequest and the active configuration in the servlet engine. The plug-in configuration files aregenerated periodically so a delay can occur between the time a change is made in the system andthe time the change is reflected in the configuration files.

The following error descriptions are symptoms of plug-in problems:

1. Servlet requests are not fulfilled. Verify the following to determinethe cause of the problem:
   - The Web server can serve HTML pages
   - The administrative console can connect to the Web server
   - The Default server is started
   - Ensure the Web server hostname and port number are identical to the ones definedin the virtual host's alias table.
   - Ensure the appropriate `.DLL` file for the Web server is present in the `<WebSphere-root>bin`
2. Pipe broken messages appear. Verify the following to determine the cause ofthe problem:
   - TCP/IP connection exists between Web server and WebSphere Application Server.
   - No process or thread failures occurred.
   - No access violations occurred.

How to debug plug-in problems

Check for errors in the following logs, and in the trace file:

- trace.log.<http server>.<date>
- <AppServer>_stderr.log
- <AppServer>_stdout.log
- <AppServer>_native.log.<date>

See files 8.3: Logs and 8.4: Trace for more information on logs and trace file.

If there are no entries in the logs or trace file, comment out the WebSphere ApplicationServer plugin in the httpd.conf file.This will help you determine if thefailure originates with WebSphere Application Server or the HTTP Server.

The WebSphere plugin property in the httpd.conf file is:

```
Load Module ibm_app_server_module
```

Restart the HTTP Server. If the Web Server initializes and runs, then WebSphere Application Serverhas a problem.

8.5.2: Servlet redirector problems

Servlet redirectors are used to separate the HTTP server from the WebSphere Application Server.There are different types of redirectors:

- Thick - servlet redirector resides on the same machine as WebSphere Administrative Server.
- Thin - servlet redirector runs on a separate machine from the WebSphere Administrative Server.
  A *thin* servlet redirector is useful in network configurations where the HTTP server is outside a firewall but WebSphere Application Server isbehind a firewall, or where WebSphere Application Server is located in the DMZ, a machine located between twofirewalls.

  > **Note:** If the Web server, application server and the server handling servlet requestsall reside on the same machine, use Open Servlet Engine(OSE), instead of a servlet redirector

Key features

The key features of servlet redirectors are:

1. Use Internet Inter-Orb Protocol (IIOP) for communication
2. Are initialized by copying the queues.properties, rules.properties,and vhosts.properties files from the WebSphere Application Server machine to the servlet redirector machine.
3. Use the following ports to transfer the properties files:
   - port 900 - bootstrap port
   - port 9000 - name services port
   - redirector listener port
4. Require the receiving RemoteSRP bean to be running on the WebSphere Application Server
5. When servlet redirectors are running, beside the ports previously listed, they alsorequire the following, additional ports:
   - Application Server listener port
   - Admin server nanny listener port
   - *Thick* servlet redirectors also require repository database connection ports

Typical problems

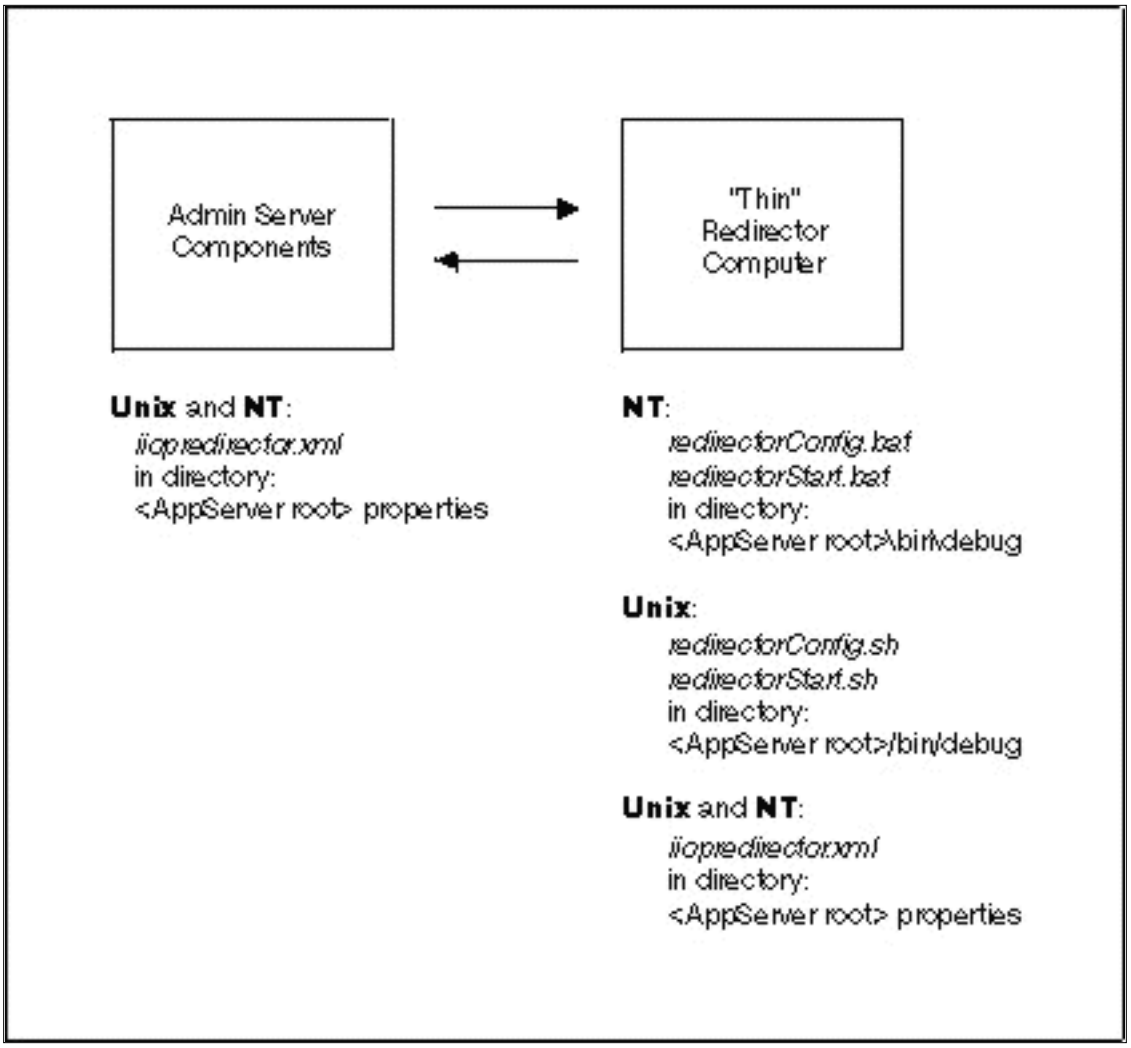
Typical problems with servlet redirectors are:

1. **Error 404** - URL not found appears on browser when accessing a servlet.
   The trace log for the HHTP server contains entry:
   Failure while locating a vhost for <server>
   This error may occur if:
   - The short (myserver) and fully qualified (myserver.mydomain.com) host names of theHTTP server are not in the virtual host alias list.
   - A port other than the standard port 80 is used and that port is not in thevirtual host alias list.
2. CORBA.COMM_FAILURE appears when running the thin servlet redirector. Thiserror may occur if:
   - Syntax error exists in batch file
   - Host name/port number is not in DNS or hosts.properties file on servlet redirectormachine
   - Host name/port number is not in DNS or hosts.properties file on WebSphere ApplicationServer machine

- -ListenerPort parameter value is not unique
- Syntax error in -ListenerPort parameter

3. Unable to initialize threads: (null) error when running the batch/script fileto configure the plug-in files.This error may occur if:
   - Some JDK other than the IBM supported JDK is in the classpath ahead of the supported JDK. To resolvethis problem, install the supported IBM JDK, and change the path and classpath to point to it first.
   - Path does not point to java.exe.
4. When starting servlet redirector with batch/script file, get error message "Errorlocating Remote SRP Home - Attribute Not Set." This error occurs because:
   - RemoteSRP bean was not added and started on WebSphere Application Server
5. Do not use servlet redirectors if you must:
   - Enable security on your thin servlet redirector machine
   - Provide the real Web browser on your thin servlet redirector machine
   - Use the Network Address Translation feature of firewalls so thatthe internal address of WebSphere Application Server is not available outside the firewall

   Reminder: Generally errors occur configuring the thin servlet redirectorfunction because required configuration files are missing. A thinservlet redirector machine requires the following shell script files (on UNIX platforms), or bat fileson Windows NT as well as the iiopredirector.xml file to enable the servlet redirector function.


Admin Server
Components
"Thin"
Redirector
Computer
Unix and **NT**:
iiopredirector.xml
in directory:
<AppServer root> properties
NT:
redirectorConfig.bat
redirectorStart.bat
in directory:
<AppServer root>\bin\debug
Unix:
redirectorConfig.sh
redirectorStart.sh
in directory:
<AppServer root>/bin/debug
Unix and **NT**:
iiopredirector.xml
in directory:
<AppServer root> properties

8.5.3: Workload management/cloning/remote administration problems

Unlike WebSphere Application Server Advanced Edition, the Standard Edition does not supportthe workload management function. WebSphere Application Server Standard Edition is limited toa single physical server. However, Standard Edition provides multiple JVMs that can be mappedto multiple virtual hosts on a single HTTP Server. Therefore multiple Web sites can still be hosted usingone Standard Edition Application Server.

Description of the workload management function and cloning

Like the Standard Edition, WebSphere Application Server Advanced Editionprovides multiple JVMs, but it also supports distributed system management for networksand clusters of Advanced Edition Application Servers. Distributed system management is anoutcome of the the Workload Management function. This function enablescloning, or creating instances of a model. Clones can be created on a single machine or across multiplemachines in a cluster. Modifying a model automatically propagates the changes to all of its clones.By administering a model, you can simultaneously administer several copies of a server (the clones).

Scalability in a workload management environment occurs in two forms:

- Horizontal scaling - the number of concurrent connected users
- Vertical scaling - the number of concurrent requests

With workload management, requests can be routed to any server on any node with thesame results. Workload management provides for failover where requests are routed to othernodes (servers) when one node fails. Failover occurs without affecting throughput or reliability.

Workload management and cloning problem descriptions

The following resources can be cloned. Select a specific type of resource to view typicalproblem scenarios.

- Application servers
- EJB containers
- Enterprise beans
- Servlet engines
- Web applications
- Servlets

Application server issues

There may be some performance impacts using cloning:

1. Make small, incremental changes to your model to minimize performance hitson the client.
2. Change your model when few clients and servers are running.
3. You do not need to reconfigure server groups to reflect an unavailable server unlessthat server will be unavailable for an extended period of time. In that case, reconfigure themodel to optimize performance.

EJB containers and enterprise beans

Homes of session beans have special information to indicate that these "homes" areworkload management

(WLM) enabled; that is, deployed on one or more servers. When a client accessesthe home of a session bean, the client is referred to the homeOfHomes on the Admin Server. The clientthen creates proxies for each clone using a specific workload management policy as for example,round robin.

Problems occur when:

- Client stubs are generated that are not WLM enabled. All stubs generated by VisualAge for Java areWLM enabled. For those stubs not created through VisualAge for Java, you must use the wlmjar file.
- Client stubs are WLM enabled but should not be. To create non-WLM stubs, use "rmic -iiop" and jarthe output files.
- You WLM enable a stateful session bean instead of its container.
- If all your requests are sent to the same server, verify that your client jar fileis WLM enabled. Use `jar -tvf` on the jar file, and verify the jar file containsboth x_Stub and x_BaseStub, not just x_Stub.

Servlet engines, Web applications and servlets

If requests are going to one server, verify that your model has more than one clone. Alsoverify that each clone has xBean deployed. This particularly applies to a *Thin Servlet Redirector*configuration since the ThinRedirector is an EJB client to the RemoteSRP bean.Also verify that the first server in the model is running. The first server must be runningfor the lookup phase to work correctly.

Remote administration

You can remotely administer WebSphere Application Server using:

- Remote administrative console
- X Windows clients on UNIX machines
- Web based WebSphere administrative console

Remote administrative console

Use can run admin console remotely using the adminclient.bat file on Windows NT, oradminclient.sh file on UNIX. See articleAdministrative models for more information on implementing a remoteadmin client.

Typical remote admin console problems are:

- The com.ibm.ejs.util.cache.FaultException error occurs in the stack trace becausethe JDK running on the client machine cannot communicate with the JDK running on the administrative servermachine. The resolution is to upgrade the backlevel JDK.
- The Network Address Translation (NAT) function in firewalls cannot be used with a remote administrative client.The internal address of the administrative server is not recognized by the administrative client outside thefirewall. No circumvention exists for this problem.

X Windows clients on UNIX machines

X windows client software can run on any platform but requires a UNIX X Windows server.(Currently the X Windows server is only available on AIX and Solaris platforms.)

Typical X Windows client problems are:

- You cannot run an administrative console remotely through the X Windows client using anauthorized, non-root account with global security enabled.The error message, FATAL Could not bind to the Administrative Server on {0}{1}, appears onthe screen when the adminclient.sh or .bat file is

executed.No circumvention exists for this problem.

Web-based administrative console

See article Web administrative console overview for informationon configuring and implementing a Web-based administrative console.

8.5.4: Installation problems

Succesful installation means that no errors occur during the install process and,more importantly, that the product runs correctly the first time you start it.

Installation and start-up problems occur for one of the following reasons:

- Database is not configured properly
- Classpath is incorrect
- Administrative server fails to start

Install options

WebSphere Application Server provides a Java ™ Graphical User Interface (GUI)install that is available on all platforms, and a native install that is available on theUNIX platforms (AIX, Solaris, HP).

> **Note:** If you used the native install option to installWebSphere Application Server on a UNIX platform, you **must** also ***uninstall***using the native uninstall option. In other words, you cannot do a nativeinstall and a Java GUI uninstall.

Follow the steps in one of the "platform specific" install guides toinstall the product. These install guides are available from the InfoCenter, inthe Selecting installation steps section.

Database configuration problems

If the database is not configured properly, installationof WebSphere Application Server will fail. If specific WebSphere components did install but thedatabase is misconfigured, the product will not run properly.

Starting WebSphere Application Server with an improperlyconfigured database will generate the following error messages and exceptions:

```
Establishing connection please wait ...  Error - could not get attributescom.ibm.ejs.util.cache.FaultException at
java.lang.Throwable<init>com.ibm.ejs.sm.client.ClientException getAttributeFailure Attributes may be
involvedcom.ibm.ejs.sm.client.RpositoryOpException could not get attributes
```

Classpath problems

The classpath provides the Java™ runtime environment for thefollowing WebSphere Application Server processes:

- Administrative service - the backend process for system management
- Administrative console - the Graphical User Interface (GUI) used for system management
- One or more application servers - each application server consists ofmultiple containers for deployment of Enterprise Java Beans (EJBs) and one servlet engine for deployment of Web applications
- Nanny service (on UNIX platforms only) - a daemon that monitorsthe administrative service. The nanny service starts the administrative serviceinitially and restarts it if it fails.

Each of these processes runs in its own Java Virtual Machine (JVM). The classpath for each process tellsthat process where to search for classes. The classpath can be set:

- In an administrative service startup script
- In an admin.config file
- With application server command line arguments
- By Web applications

Classpath properties

Each process has an associated set of properties ("Java-speak" for environment variables).These properties are defined in the admin.config file that is located in directory:

```
<WebSphere root> bin
```

The applicable properties in admin.config are:

- com.ibm.ejs.sm.util.process.Nanny.maxtries
- com.ibm.ejs.sm.adminserver.classpath
- com.ibm.ejs.sm.util.process.Nanny.path
- com.ibm.ws.jdk.path

The classpath settings in the admin.config file apply to the administrativeservice, and they are also inherited by all other WebSphere Application Server processes.

For more information on these and other WebSphere Application Server properties, see file,6: Administer applications.

Classpath failures

Typical classpath failures are:

1. When a servlet class is missing from a Web application classpath, the following errors occur:
   - In a browser window, the browser displays error ***Error 500*** with message,"Failed to load target servlet [snoop]."
   - Browser stack trace and <AppServerName>_stderr.log show java.lang.ClassNotFoundException
   - <AppServerName>_stdout.log shows javax.servlet.ServletException
2. When utility classes, such as dates.class or time.class, are ina Web application's classpath, the following errors occur:
   - Browser shows error messagejava.lang.VerifyError
   - Verbose JVM output written to the <AppServerName>_stderr.log shows java.lang.VerifyError: com/bcs/jsftest/test

     Note: A WebSphere Application Server problem exists onthe Windows NT platform which prevents the stderr buffer from being flushed until the applicationserver is stopped. No circumvention for this problem is available at this time.
3. When classes use Java Native Interface (JNI), the following errors occur:
   - <AppServerName>_stdout.log shows java.lang.UnsatisfiedLinkError

   To resolve the problem, do the following:
   - Ensure the shared libraries are available in the path statement on Windows NT.On UNIX, make sure the LD_LIBRARY_PATH is defined in file startupServer.sh.
   - Ensure that the property defined in com.ibm.ejs.sm.adminserver.classpath, in the admin.configfile, includes classes that make JNI calls into shared libraries.

Administrative server problems

Successfully starting the administrative server not only indicates a successful install of WebSphere ApplicationServer, but it also means the following tasks were completed:

- System management repository tables were created in the database.
- Nodes and host aliases were created in the repository tables with xml.
- Default repository tables were created with xml.

Therefore, when the administrative server fails to start, it also means the installation of WebSphere Application Serveris incomplete.

Administrative server start failures

The administrative server fails to start for the following reasons:

1. The port is in use. See the port problems section for more information.
2. Another administrative server is running. The administrative server service in the Windows NT control panel orthe startupServer.sh script on UNIX, is the same service/process as the onestarted through WebSphere\Appserver\bin\debug\adminserver.bat file on Windows NT oradminserver.sh on UNIX.
3. The WebSphere Application Server database repository, (WAS on DB2 orORCL on Oracle), is not created.The first time you start the administrative server process, itattempts to create the default configuration in the WAS or ORCLdatabase. You will see a 2140 error message if the database is not created.
4. Connection to DB2 or Oracle fails. This also shows up as a 2140 error message.Ensure DB2 is running. Verify the connectionto the WAS database is successful.
   To test the DB2 connection, from a DB2 command window, type:

   ```
   DB2 connect to was
   ```

   If you cannot connect to DB2, verify the following:
   - Ensure the right level of codeis installed on the WebSphere Application Server machine
   - For a remote repository, ensure the DB2 client is configured properly topoint to DB2 server for the WAS database.

   Perform the same tests for Oracle.

5. User ID does not have proper authority or access:
    - To ensure proper authority, follow the database configuration stepsin the install guides.
    - In the UNIX environment, log on as ***root*** to start AdminServer.
    - In the Windows NT environment, verify the following conditions are true:
        - User is logged in as an administrative user
        - User name in security panel is correct
        - User is part of the administrator's group.
        - The Administrative server is registered as a service to NT. To manually add theadministrative server as a service, from a command prompt, enter:
          <WebSphere\AppServer>\bin\adminservice.exeinstall <WebSphere\AppServer>\bin\admin.config <HostName>\<User> <Password>
        - User ID has proper rights to start the administrative server. If using a domain ID, start the administrative server with a local IDto see if the domain is the problem. To check a user's rights:
            1. From ***Start*** > Programs > Administrative Tools > User Manager
            2. Select ***Policies*** > ***User Rights***
            3. Check ***Show Advanced User Rights*** checkbox in lower left corner
            4. Add the following rights to the user ID:
                - Log on as a service
                - Act as part of the operating system
        - If you change the Windows NT user ID/password but WebSphere Application Server isnot updated, then the administrative server startup will fail.
          Update the user ID/password in the following areas:
            - In Windows NT services for the IBM WebSphere Administrative Server service:
                1. From ***Start*** > Settings > Control Panel, double click ***Services***
                2. Select ***IBM WebSphere Administrative Server***
                3. Click ***Startup***
                4. Change the user ID/password under this account
            - in admin.config (if the DB2 userid/password also changed)

Port problems

WebSphere Application Server will fail to start if certain ports are in use. Typical port problem descriptions follow:

1. When the bootstrap port is in use,you may see the following error when starting WebSphere:

   ```
   009.765.6005c5b F Nameserver Failed to start the Bootstrap server
   org.omg.CORBA.INTERNAL: minor code: 8 completed: No
   ```

   This error is similar to the *Port 9000 in use* error when starting WebSphere Application Server.

   To fix the problem, change the bootstrap port (the default is 900) in file, ***admin.config***,using property name:

   `com.ibm.ejs.sm.adminServer.bootstrapPort`

   If this property does not exist in file ***admin.config***, add it.

2. Port 9000 is the default port of the Admin Server location service daemon.Port 9000 is also used by many system resources including AIX X-windows manager.If you see error message,

   `Port 9000 in use-select another port`

   when executing the ./startupServer.sh command on AIX,the Admin Server process cannot start because port 9000 is in use.You can change the port the location service daemon listens on by:

    - specifying ***-lsdPort*** option on the admin server command line
    - setting ***com.ibm.ejs.sm.adminServer.lsdPort*** property in the *admin.config* file locatedin directory **<WAS_ROOT>\bin** on

Windows NT and**<WAS_ROOT>/bin** on UNIX.

8.6: Diagnosing configuration and installation problems

WebSphere Application Server uses a database to store and share configuration information acrossnodes. Problems configuring the database are described in the installation problemssection.

Generally, if the database is not configured properly, theWebSphere Application Server installation process will fail.Configuration problems occur after the product is installed.

The following table describes common configuration problems. Select an entry for moreinformation.

Problem description	Cause of problem
Cannot retrieve data for a specific session	Incorrect use of the database as a session store: • Datasource incorrectly created or configured • EJSPersistenceException occurs
java.io.InvalidClassException occurs after bean is modified	Serialized descriptor must be recreated
Servlet requests are not satisfied	Web server problems: • Plug-in failure • Virtual host configuration incorrect
Error 404 - URL not found occurs when accessing a servlet	HTTP Server hostname or port problem
CORBA.COMM_FAILURE appears when running thin servlet redirector	Servlet redirector configuration problem
Error message - Error locating RemoteSRP Home - Attribute Not Set	Receiving RemoteSRP bean is not running
Error 500 - Failed to load target servlet	Servlet missing from Web application classpath
FATAL - Could not bind to the administrative server error	Remote administration failure
Changes to clones not visible	Workload management/cloning failures
Nanny process fails to start administrative server	Verify all installation steps were successful
Administrative server fails to start	Generally an installation setup problem

8.6.1: ORB-related Minor Codes

This document provides explanations of the minor error codes used by the WebSphere Application Server Advanced Edition Java ORB. These minor codes are not CORBA-compliant. CORBA-compliant minor codes usually begin with an OMG-assigned identification code, which consists of the vendor ID followed and digits that identify the minor code. However, the Java ORB minor codes do not contain the vendor ID.

Minor codes are associated with CORBA exceptions and provide greater detail about the errors that can occur. There is not a one-to-one mapping of exception names to minor codes. Instead, a minor code can be associated with several exception names. A minor code message can have different meaning s depending on the excepton that was thrown.

Minor codes are scoped to System Exceptions in the range 0 to 4095. A minor code ID must be a unique number within the scope for each System Exception, but there is no restriction that minor codes be unique across all System Exceptions.

The following table lists the System Exceptions and the corresponding minor error codes, where:

- Minor code: The minor error code
- Static variable: The name of the static variable for the minor error code
- Explanation: A description of the problem that caused the error
- User response: Actions to resolve the problem

org.omg.CORBA.BAD_PARAM

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NULL_PARAM
- Explanation: A parameter with a value of NULL was received. The parameter is not valid.
- User response: Ensure that parameters are initialized correctly.

org.omg.CORBA.COMM_FAILURE

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.CONNECT_FAILURE
- Explanation: The ORB could not establish a connection to the server on the host and port that was identified by the object reference.
- User response: Ensure that the server is running and listening on the designated host and port.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.CONN_CLOSE_REBIND
- Explanation: A client request could not be processed, because the server had notified the client to close the connection and a new connection could not be established.
- User response: Ensure that the server is running and try the request again.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.WRITE_ERROR_SEND
- Explanation: An error was encountered while writing the request to the output stream.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.GET_PROPERTIES_ERROR
- Explanation: An exception was encountered while reading the initial services from a URL.
- User response: Ensure that the initial services URL is valid.

- Minor code: 6
- Static variable: com.sun.rmi.util.MinorCodes.INVOKE_ERROR
- Explanation: The ORB was unable to successfully connect to the server after several attempts.
- User response: Ensure that the server is running.

org.omg.CORBA.DATA_CONVERSION

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.BAD_HEX_DIGIT
- Explanation: The object reference in string format contains at least one hexadecimal character that is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method on the ORB.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_STRINGIFIED_IOR_LEN
- Explanation: The length of the string-formatted object reference is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method on the ORB.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.BAD_STRINGIFIED_IOR
- Explanation: The string-formatted object reference is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method on the ORB.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.BAD_MODIFIER
- Explanation: The initial reference could not be resolved, because the host or the port is not valid or was not specified.
- User response: Specify the correct host and port.
- Minor code: 5
- Static variable: com.sun.rmi.util.MinorCodes.CODESET_INCOMPATIBLE
- Explanation: While processing the service context code sets for a request, an incompatible code set was encountered.

org.omg.CORBA.INTERNAL

- Minor code: 8
- Static variable: com.sun.rmi.util.MinorCodes.CREATE_LISTENER_FAILED
- Explanation: The ORB could not establish a listener thread on the port identified by the object reference. The port was already in use or there was an error in creating the daemon thread.
- Minor code: 9
- Static variable: com.sun.rmi.util.MinorCodes.BAD_LOCATE_REQUEST_STATUS
- Explanation: The locator performed a locate request for an object reference and returned a locate reply with a status that is not valid.
- Minor code: 10
- Static variable: com.sun.rmi.util.MinorCodes.STRINGIFY_WRITE_ERROR
- Explanation: An exception was encountered while attempting to create a string-formatted object

reference.

org.omg.CORBA.INV_OBJREF

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NO_PROFILE_PRESENT
- Explanation: The object reference does not contain a profile.
- User response: The current object reference is not valid. Obtain a valid object reference from the object supplier.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_CODE_SET
- Explanation: An unsupported code set or a code set that is not valid was used to write the data to the input stream.
- User response: Use a Unicode or ASCII code set.

org.omg.CORBA.MARSHAL

- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.READ_OBJECT_EXCEPTION
- Explanation: An error was encountered while trying to read and convert a marshalled object reference into an in-memory object.
- User response: Ensure that the object (passed as a parameter) is in one of the locations identified by the system CLASSPATH environment variable.
- Minor code: 6
- Static variable: com.sun.rmi.util.MinorCodes.CHARACTER_OUTOFRANGE
- Explanation: While marshalling or unmarshalling an object, a character that is not compliant with ISO Latin-1 (8859.1) was encountered. The character is not in the range 0 to 255.

org.omg.CORBA.NO_IMPLEMENT

- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.GETINTERFACE_NOT_IMPLEMENTED
- Explanation: The get_interface method is not implemented on the server.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.SEND_DEFERRED_NOTIMPLEMENTED
- Explanation: Deferred sends are not supported by the ORB.

org.omg.CORBA.OBJ_ADAPTER

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_DISPATCH
- Explanation: The object reference could not be dispatached to the server, because an object adapter that matches the object key could not be found.
- User response: Ensure that the object server still services the requested object.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_LOOKUP
- Explanation: The requested object could not be located, because an object adapter that matches the adapter that matches the object key could not be found.
- User response: Ensure the object server that processes the locate requests still services the requested object.

- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_CREATE_DEFAULT_SERVER
- Explanation: The ORB was unable to create the default object adapter for an object in the server that processes the actual method call.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.ORB_CONNECT_ERROR
- Explanation: An error was encountered while trying to connect to an object in the server that processes the actual method call.
- User response:

org.omg.CORBA.OBJECT_NOT_EXIST

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.LOCATE_UNKNOWN_OBJECT
- Explanation: A locate request was performed and the response indicated that the object is not known to the locator.
- User response: Ensure that the locator that processes the locate requests still services the requested object.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_SERVER_ID
- Explanation: The server ID of the server that received the request does not match the server ID of the request object reference. The server that originally served the object is no longer identified by that server ID.
- User response: Obtain a new object reference for the object from the server that is now servicing that object.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.BAD_IMPLID
- Explanation: The implementation ID (identified by the object reference) does not match any implementation on the server.
- User response: Obtain a new object reference for the object from the server that is now servicing that object.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.BAD_SKELETON
- Explanation: A skeleton that matches the object reference (identified by the object key) could not be found on the server.
- User response:
- Minor code: 5
- Static variable: com.sun.rmi.util.MinorCodes.SERVANT_NOT_FOUND
- Explanation: The object adapter identified by the object key within the object reference could not locate the servant (an object on the server) to process the object request.
- User response: Ensure that the servant is known to the object adapter.

org.omg.CORBA.UNKNOWN

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.UNKNOWN_CORBA_EXC
- Explanation: The server threw an unknown user exception.
- User response: Ensure that all user exceptions that can be thrown are declared on the throws clause of the method.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.RUNTIMEEXCEPTION
- Explanation: The server encountered an unknown application error.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.UNKNOWN_SERVER_ERROR
- Explanation: The server threw an unknown exception.

8.7: Using application level facilities

WebSphere Application Server Standard Editiononly supports Web applications, not enterprise beans. WebSphere Application ServerAdvanced Edition supports both Web applications and enterprisebeans.

For more information on enterprise beans and Web applications,see the file on developing applications.

Tools that are specifically designed to debug application, servlet and EJB problems include OLTand Distributed Debugger.

Typical application and EJB problems are:

- Invoking a servlet from a browser window
- A modified servlet is not reloaded
- Incorrectly using of databases as a session store
- Various exceptions running enterprise beans

Invoking a servlet by its URL

The following example describes what you should enter in a browser window toinvoke a servlet. Errors occur when you fail to include the webappdirectory in the path.

http://server_machine/webapp/examples/showCfg

The components of the URL are:

server_machine	webapp	examples	showCfg
Name of the application server computer	Virtual directory of the Web application loader. Do not create a webapp directory. This directory is defined for you by WebSphere Application Server.For more information on *webapp*, see the file on the programming model and environment.	Application Web path This is a default WebSphere servlet Web path.You can create a directory by any name as long as it is defined in the Web application'scategory.	Servlet URL, not the name of the code. In this example, the actual Servlet class name isServletEngineConfigDumper.

The URL illustrated above is the URL for showCfg, one of the default servlets shipped with WebSphereApplication Server.

Reloading servlets

In earlier versions of WebSphere Application Server, specific reload directories in the reload processhad to be defined. Currently, the only reload requirement is to store servlet classesin the Web application category. That is, ensure all your servlets arehandled in the context of the Web application loader. After you update your servlets, the Webapplication loader will automatically reload them for you.

If your servlet classes are installed in the context of the Web application loader, butare not being reloaded, ensure the Auto Reload property is set to true.Follow these steps to check the setting of the Auto Reload property:

1. From the WebSphere Administration Console, select your application (or default_app if you storedyour servlets in the default directory structure).
2. From the Web Application:default_app panel, select the **Advanced** tab.
3. Verify that the Auto Reload is set to true.

Incorrect use of a database as a session store

WebSphere Application Server makes JDBC calls, using a predefined JDBC driver, to communicatewith a database. Both the JDBC driver and datasources must be configured using the administrative console.

The following errors occur if a datasource is misconfigured or does not exist:

- The browser window displays Error 500 with the message: `java.lang.NullPointerException`
- The <App_Server>.stderr.log displays the message: `javax.naming.NameNotFoundException: jdbc/xxx`
- The <App_Server>.stdout.log displays the message: `Failure while creating connection COM.ibm.db2.jdbc.app.DB2Exception: [IBM] [CLI Driver] SQL1013NThe database alias name or database name "SAMPLE" could not be found. SQLSTATE=42705`

Database connectivity problems cause persistence exceptions. AnEJSPersistenceException error may indicate JDBC or connection problems:

1. An invalid JDBC driver will prevent access to jar and class files
2. Review the SQLSTATE:`COM.ibm.db2.jdbc.app.DB2Exception: [IBM][CLI Driver]SQL1224N A database agent could not be started...SQLSTATE=55032...`The SQLSTATE code of 55032 indicates the system is out of connections.

   Note: Not using connection pooling causes most problemsfor BMP type EJBs. Common symptoms include:

   - Performance problems connecting to the database
   - Running out of connections

   To resolve the problem:

   1. Increase the number of connections permitted by DB2 or Oracle.
   2. On AIX, catalog the database as if it were remote.
   3. Ensure you close connections when programming exceptions occur.
   4. Verify that connections obtained in one method,are returned to the pool via close().
   5. Verify that your application does not try to access pre-empted connections (idle connectionsthat are now used by other resources).
3. A database init failure could indicate the database does not

exist:`com.ibm.ejs.persistence.EJSPersistenceException: Database init failure:` Nested exception is:`COM.ibm.db2.jdbc.app.DB2Exception: [IBM][CLI Driver]SQL1013N The database alias name or database name "YYY" could not be found...SQLSTATE=42705...` The SQLSTATE code of 42705 indicates the database does not exist or the server cannotconnect to it.

Various exceptions running EJBs

EJB's can be classified as:

- Session beans - extensions of client applications without persistence
- Entity beans - a type of BMP (Bean Managed Persistence) where implementer manages persistence
- Entity beans - a type of CMP (Container Managed Persistence) where framework manages persistence

See tbe file about writing enterprise beans, for a description of EJB components andfunctions.

A summary of EJB failures

1. If you modify a bean and then see the following exception:`java.io.InvalidClassException: xpackage.xBean;Local class not compatible: stream classdesc serialVersionUID=54545...Local class serialVersionUID=589090...` the deployment descriptor (also known as the serialized descriptor) is missing and must be recreated.

   To recreate the deployment descriptor follow these steps:

   1. Run the JETACE tool from the GUI or in line mode
   2. Use the jetace.bat file on Windows NT or jetace.sh on UNIX to run theJETACE tool (com.ibm.ejb.jetjar.JetAceMain) in line mode
   3. The output jar file is the EJB jar file that you will deploy in the WebSphere AdministrativeConsole.
   4. The EJB jar file will contain your EJB classes and a deployment descriptor (.ser) file.
2. If your EJBs use JNI (Java Native Interface), loaded libraries should be added to:
   - Path on Windows NT
   - Whatever is defined for the LD_LIBRARY_PATH property in file startupServer.sh on UNIX

   Missing libraries will cause the following error:`java.lang.UnsatisfiedLinkError: no <missing DLL> in shared library pathat java.lang.Runtime.loadLibrary(Runtime.java:440)at java.lang.System.loadLibrary(System.java:569)` You must stop and restart the server after the problem is resolved.

3. If your client application or servlet fails during operation,
   JNDI ctxt.lookup("myinterface")
   review the generated exception:
   - When you experience the exception, Cannot instantiate class:`com.ibm.ejs.ns.jndi.CNnitialContextFactory`,the classpath is incorrect. Add ibmwebas, ejs and ujc jar files and retry.
   - When you experience the exception, COMM_FAILURE, verify the administrative server host is running andaccessible by entering the command:`telnet <host> 900` If telnet fails, either the host is not listening on port 900 or a firewall is not permittingthe connection (check the filters settings in the firewall).

If telnet hangs, the connection is successful, but a stale home reference is left over in thename space; that is, no server implemented that bean. Redeploy the EJB jar file ona server.

- When you experience the error, java.lang.NoClassDefFoundError: myejbHome,add the client EJB jar file to the client classpath.

8.7.1: ORB-related minor codes

This document provides explanations of the minor error codes used by the WebSphere Application Server Advanced EditionJava ORB. These minor codes are not CORBA-compliant. CORBA-compliant minor codes usually begin with an OMG-assignedidentification code, which consists of the vendor ID and digits that identify the minor code. However, the Java ORB minorcodes do not contain the vendor ID.

Minor codes are associated with CORBA exceptions and provide greater detail about the errors that can occur. There isnot a one-to-one mapping of exception names to minor codes. Instead, a minor code can be associated with severalexception names. A minor code message can have different meanings depending on the exception that was thrown.

Minor codes are scoped to system exceptions in the range 0 to 4095. A minor code ID must be a unique number withinthe scope for each system exception, but there is no restriction that minor codes be unique across all system exceptions.

The following table lists the system exceptions and the corresponding minor error codes, where:

- Minor code: The minor error code
- Static variable: The name of the static variable for the minor error code
- Explanation: A description of the problem that caused the error
- User response: Actions to resolve the problem

org.omg.CORBA.BAD_PARAM

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NULL_PARAM
- Explanation: A parameter with a value of NULL was received. The parameter is not valid.
- User response: Ensure that parameters are initialized correctly.

org.omg.CORBA.COMM_FAILURE

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.CONNECT_FAILURE
- Explanation: The ORB could not establish a connection to the server on the host and port that was identified by the object reference.
- User response: Ensure that the server is running and listening on the designated host and port.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.CONN_CLOSE_REBIND
- Explanation: A client request could not be processed, because the server had notified the client to close the connection and a new connection could not be established.
- User response: Ensure that the server is running and try the request again.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.WRITE_ERROR_SEND
- Explanation: An error was encountered while writing the request to the output stream.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.GET_PROPERTIES_ERROR
- Explanation: An exception was encountered while reading the initial services from a URL.
- User response: Ensure that the initial services URL is valid.

- Minor code: 6
- Static variable: com.sun.rmi.util.MinorCodes.INVOKE_ERROR
- Explanation: The ORB was unable to successfully connect to the server after several attempts.
- User response: Ensure that the server is running.

org.omg.CORBA.DATA_CONVERSION

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.BAD_HEX_DIGIT
- Explanation: The object reference in string format contains at least one hexadecimal character that is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method on the ORB.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_STRINGIFIED_IOR_LEN
- Explanation: The length of the string-formatted object reference is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method onthe ORB.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.BAD_STRINGIFIED_IOR
- Explanation: The string-formatted object reference is not valid.
- User response: Obtain the original object reference and reformat it as a string using the object_to_string method onthe ORB.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.BAD_MODIFIER
- Explanation: The initial reference could not be resolved, because the host or the port is not valid or was notspecified.
- User response: Specify the correct host and port.
- Minor code: 5
- Static variable: com.sun.rmi.util.MinorCodes.CODESET_INCOMPATIBLE
- Explanation: While processing the service context code sets for a request, an incompatible code set was encountered.

org.omg.CORBA.INTERNAL

- Minor code: 8
- Static variable: com.sun.rmi.util.MinorCodes.CREATE_LISTENER_FAILED
- Explanation: The ORB could not establish a listener thread on the port identified by the object reference. The portwas already in use or there was an error in creating the daemon thread.
- Minor code: 9
- Static variable: com.sun.rmi.util.MinorCodes.BAD_LOCATE_REQUEST_STATUS
- Explanation: The locator performed a locate request for an object reference and returned a locate reply with a statusthat is not valid.
- Minor code: 10
- Static variable: com.sun.rmi.util.MinorCodes.STRINGIFY_WRITE_ERROR
- Explanation: An exception was encountered while attempting to create a string-formatted object

reference.

org.omg.CORBA.INV_OBJREF

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NO_PROFILE_PRESENT
- Explanation: The object reference does not contain a profile.
- User response: The current object reference is not valid. Obtain a valid object reference from the object supplier.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_CODE_SET
- Explanation: An unsupported code set or a code set that is not valid was used to write the data to the input stream.
- User response: Use a Unicode or ASCII code set.

org.omg.CORBA.MARSHAL

- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.READ_OBJECT_EXCEPTION
- Explanation: An error was encountered while trying to read and convert a marshalled object reference into an in-memory object.
- User response: Ensure that the object (passed as a parameter) is in one of the locations identified by the system CLASSPATH environment variable.
- Minor code: 6
- Static variable: com.sun.rmi.util.MinorCodes.CHARACTER_OUTOFRANGE
- Explanation: While marshalling or unmarshalling an object, a character that is not compliant with ISO Latin-1 (8859.1) was encountered. The character is not in the range 0 to 255.

org.omg.CORBA.NO_IMPLEMENT

- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.GETINTERFACE_NOT_IMPLEMENTED
- Explanation: The get_interface method is not implemented on the server.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.SEND_DEFERRED_NOTIMPLEMENTED
- Explanation: Deferred sends are not supported by the ORB.

org.omg.CORBA.OBJ_ADAPTER

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_DISPATCH
- Explanation: The object reference could not be dispatached to the server, because an object adapter that matches the object key could not be found.
- User response: Ensure that the object server still services the requested object.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_LOOKUP
- Explanation: The requested object could not be located, because an object adapter that matches the adapter thatmatches the object key could not be found.
- User response: Ensure that the object server that processes the locate requests still services the requested object.

- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.NO_SERVER_SC_IN_CREATE_DEFAULT_SERVER
- Explanation: The ORB was unable to create the default object adapter for an object in the server that processes the actual method call.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.ORB_CONNECT_ERROR
- Explanation: An error was encountered while trying to connect to an object in the server that processes the actual method call.
- User response:

org.omg.CORBA.OBJECT_NOT_EXIST

- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.LOCATE_UNKNOWN_OBJECT
- Explanation: A locate request was performed and the response indicated that the object is not known to the locator.
- User response: Ensure that the locator that processes the locate requests still services the requested object.
- Minor code: 2
- Static variable: com.sun.rmi.util.MinorCodes.BAD_SERVER_ID
- Explanation: The server ID of the server that received the request does not match the server ID of the request objectreference. The server that originally served the object is no longer identified by that server ID.
- User response: Obtain a new object reference for the object from the server that is now servicing that object.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.BAD_IMPLID
- Explanation: The implementation ID (identified by the object reference) does not match any implementation on theserver.
- User response: Obtain a new object reference for the object from the server that is now servicing that object.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.BAD_SKELETON
- Explanation: A skeleton that matches the object reference (identified by the object key) could not be found on the server.
- User response:
- Minor code: 5
- Static variable: com.sun.rmi.util.MinorCodes.SERVANT_NOT_FOUND
- Explanation: The object adapter identified by the object key within the object reference could not locate the servant (an object on the server) to process the object request.
- User response: Ensure that the servant is known to the object adapter.

org.omg.CORBA.UNKNOWN

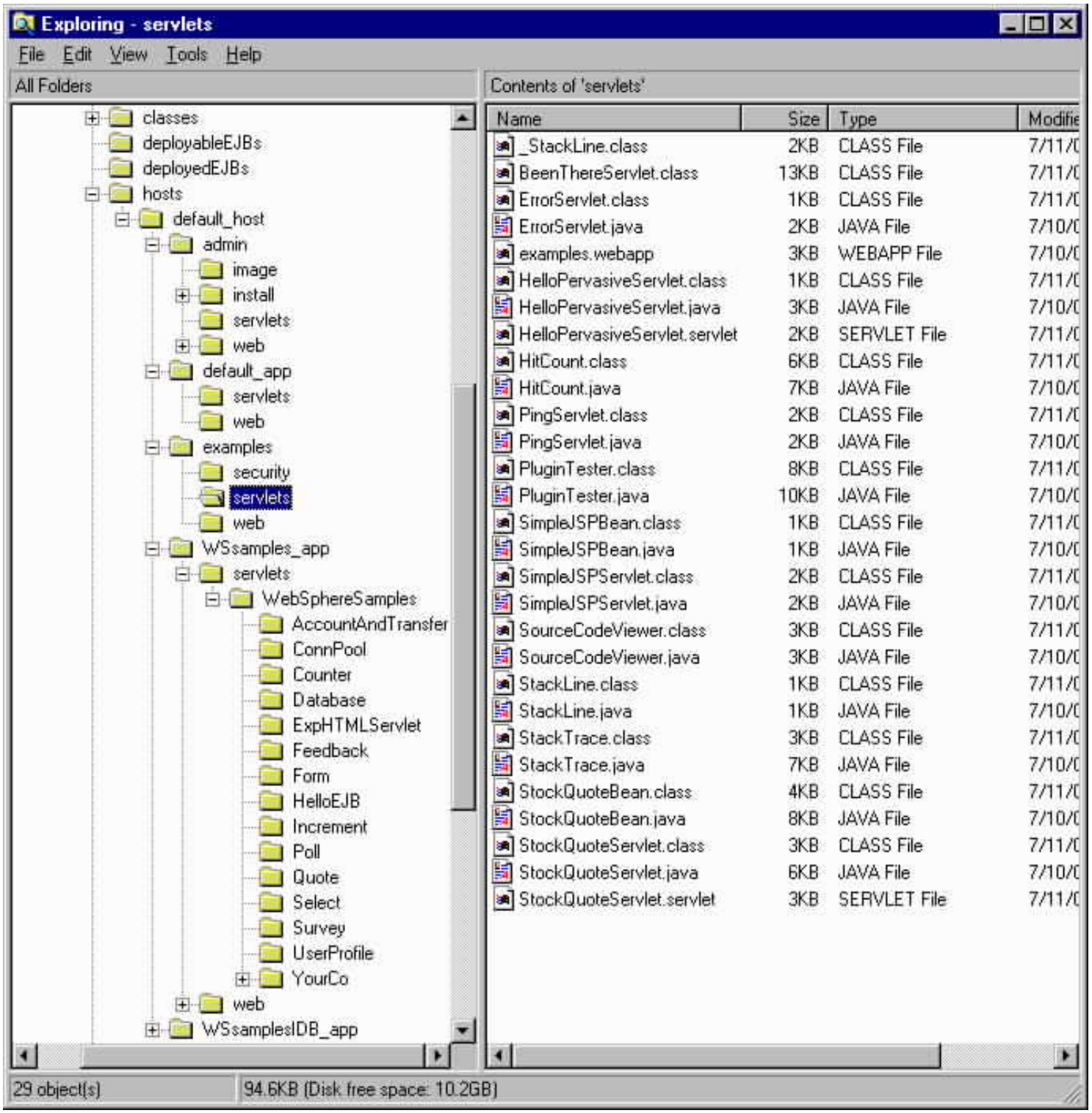
- Minor code: 1
- Static variable: com.sun.rmi.util.MinorCodes.UNKNOWN_CORBA_EXC
- Explanation: The server threw an unknown user exception.
- User response: Ensure that all user exceptions that can be thrown are declared on the throws clause of the method.
- Minor code: 3
- Static variable: com.sun.rmi.util.MinorCodes.RUNTIMEEXCEPTION
- Explanation: The server encountered an unknown application error.
- Minor code: 4
- Static variable: com.sun.rmi.util.MinorCodes.UNKNOWN_SERVER_ERROR
- Explanation: The server threw an unknown exception.

8.8: Using internal tools

You can use WebSphere Application Server servlets and internal tools to help diagnoseproblems.

Servlets

View file samples.html for information on sample serlvetsshipped with the product. Most WebSphere Application Server servletsare located in the examples directory:



The following table describes servlets that can be used as debug tools:

servlet	location	description
Hit Count	<WebSphere\AppServer>\hosts\default_host\examples\	Verifies correct implementation of Servlets, JSPs, EJBs, and HTTP Session.
Snoop Servlet	<WebSphere\AppServer>\hosts\default_host\default_app\servlets\Snoop.class	Useful for examining request parameters coming from the clientand for verifying the operation of the servlet engine.
ShowCfg	<WebSphere\AppServer>\hosts\default_host\examples\	Useful for validating the configuration of the system.
BeenThere	<WebSphere\AppServer>\hosts\default_host\examples\web\beenthere.html	Useful for demonstrating and testing session persistance.

Internal tools

The available internal tools apply to specific operations. For example, thejdbctest.java™ tool tests JDK™ settings and database access.

See theWebSphere ProblemDetermination Tools website for detailed information about these tools. Thewebsite also offers you the opportunity to add ideas about tools and add a new tool. Check the websiteperiodically for updates.

tool	description
Jdbctest.java>	Tests JDK™ settings and database connectivity
Java™ Name Tree Browser	Displays elements in WebSphere Application Server name space
Java™ Socket Level Trace	Describes ORB communication problems over heterogeneous networks via IIOP
DrAdmin trace function	Dumps the thread stacks in a server
OLT	Object level trace
Distributed Debugger	Debugs application level problems
NT Resource Analyzer	Provides monitoring and tuning support to enhance performance

8.8.1: Using the Log Analyzer for Advanced Edition

The Log Analyzer takes one or more activity logs, mergesall of the data, and displays the entries. Based on its "symptomdatabase," it analyzes and interpretsthe error conditions in the log entries to help you debug problems.The Advanced Edition Log Analyzer has a special feature enabling itto download the latest symptom database from the IBM Web site. See"About the Log Analyzer" for details.

- About the Log Analyzer
- Obtaining the Log Analyzer
- About the activity log
- Using the Log Analyzer
- Related tasks:
  - Set the maximum activity.log size
  - Changing the port of the logging service
  - View an activity.log on a remote machine

About the Log Analyzer

A Log Analyzer quite similar to the one available for use with IBMWebSphere Application Server is available with IBM Component Broker, partof the Enterprise Edition of IBM WebSphere Application Server.

To learn about the Component Broker Log Analyzer, see the Component Brokerproblem determination guide (currently, Chapter 11). You can view thedocument on the IBM Web site without having to obtain Enterprise Edition:

`ftp://ftp.software.ibm.com/software/websphere/info/appserv/v35/ee/cbprbdet.pdf`

The main differences between the Log Analyzer available withWebSphere Application Server and the Component Broker Log Analyzerare the following:

- The Log Analyzer for Advanced Edition is capable of downloadingthe latest symptom database (bin/symptomdb.xml) from the IBM support site. Usethe file -> Update Database -> Adv Symptom Database option in theLog Analyzer interface to take advantage of this feature.
- The functions for ORB trace formatting, minor codes, message IDs and GPFare **not** supported for Advanced Edition
- The script for starting the Log Analyzer is in a different location(see below for instructions)
- The default directory for opening logs for Advanced Edition is thelogs directory. For Enterprise Edition, it is the service directory.

Obtaining the Log Analyzer

The Log Analyzer is available separately from IBM WebSphere Application Server,starting on or shortly after the release of Fix Pack 2 (Version 3.5.2) forIBM WebSphere Application Server.

To obtain the Log Analyzer, including its installation instructions, visitthe IBM WebSphere Application Server Tools site:

`http://www.ibm.com/software/webservers/appserv/tools.html`

About the activity log

Recall, the Log Analyzer collects messages from various product componentsand places them in a shared file. The file is abinary file located at:

`product_installation_root``/logs/activity.log`

The activity.log cannot be easily viewed using a texteditor. The Log Analyzer is the tool for viewing the file.

Using the Log Analyzer

To view the activity.log using the Log Analyzer:

1. Change directory to:
   `product_installation_root``/bin`
2. Run the waslogbr script file, which is called:
   - waslogbr.bat on Windows NT
   - waslogbr.sh on UNIX systems

   It needs to be run from the bin directory cited above.

   This will start the Log Analyzer graphical interface.
3. In the interface:
   1. Select File-> Open.
   2. Navigate to the directory containing the activity.log file.
   3. Select the activity.log file.
   4. Select Open.

Related tasks

In the course of using the Log Analyzer, you might need to perform thefollowing tasks.

Setting the maximum activity.log file size

The activity.log file grows to a predetermined size, then wraps. The default size is1 Megabyte (MB).

To change the log size:

1. Open the properties file in a text editor:
   product_installation_root`/properties/logging.properties`
2. For the SHARED_LOG_LENGTH property, specify the value you would like, in Kilobytes (KB).

   If an invalid size is entered, the default size is used.

The size change will take effect at the next server startup.

Syntax example:

- To change the log size to 2MB, enter in the line:
  `SHARED_LOG_LENGTH=2048`
  without any spaces in it.

Changing the port of the logging service

The logging service starts automatically at server startup. It requires theuse of a dedicated port. The default port is 1707.

To change the port value:

1. Stop all application servers and the WebSphere administrative server. (Ifyou do not stop a server, it will not pick up the property changeuntil it is stopped and started again).
2. Open the properties file in a text editor:
   product_installation_root`/properties/logging.properties`
3. For the SHARED_LOG_LOCK_PORT property, specify the value you would like.
4. Start the application and administrative servers that you stopped.

Syntax example:

- To change the port to 1708, specify:
  `SHARED_LOG_LOCK_PORT=1708`

If the port is in use by another application, the logging service might notbe able to start or might not function correctly. The activity.log file willnot be created or updated correctly. See the information about how to tell whether a port is currently allocated to anotherapplication.

To diagnose a port conflict, perform these heuristic checks:

- Check to see if the activity.log file has been created, and check the timestamp of the file.
- Check these files:
  product_installation_root`/<server_name>_stderr.log`*product_installation_root*`/logs/adminserver_stderr.log`
  Note: The above paths are the default locations of the files. The administratormight have configured different locations.

  Look for a stack trace such as the following:

```
java.lang.Exception: Unable to obtain Shared Log Lock on port1707at
com.ibm.ejs.ras.SharedLogBase.acquireHostLock(SharedLogBase.java:187)at
com.ibm.ejs.ras.SharedLogWriter.<init>(SharedLogWriter.java:130)at
com.ibm.ejs.ras.SharedLogWriter.getInstance(SharedLogWriter.java:100)at
com.ibm.ejs.ras.Tr.initialize(Tr.java:241)at
com.ibm.ejs.sm.server.ManagedServer.main(ManagedServer.java:121)
```

Viewing an activity.log file in the absence of a GUI.

The Log Analyzer cannot be used to view remote files. If the operating systemon which you are running WebSphere Application Server does not support the use of agraphical interface, then transfer the file in binary to the systemon which you are running the WebSphere Java administrative console. Use the LogAnalyzer tool there.

In cases in which transferring the file is impractical or inconvenient, analternate tool named "showlog" is provided for viewing the activity.log file:

1. Change directory to:
   product_installation_root`/bin`
2. Run the showlog tool with no parameters to display the usage instructions:
   - On Windows NT, run showlog.bat
   - On UNIX systems, run showlog.sh

Syntax examples:

- To direct the activity log contents to stdout, use the invocation:

```
showlog activity.log
```

- To dump the activity.log to a text file that can beviewed using a text editor, use the invocation:

```
showlog activity.log textFileName
```

8.9: Thread dumps

This section introduces the concept of thread dumps in WebSphere Application Server.

A thread represents a work item or task, such as a servlet request. Java processes areusually multi-threaded. This means there can be many tasks occurring simultaneously(i.e. multi-threading)within one JVM (Java Virtual Machine) process. Therefore, understanding what is occurring within a JVMprocess is means obtaining information about all the different threads that are defined within the process.

What is a thread dump?

There are two types of thread dumps that could appear when running Java programs:

- System dump
- Java dump

System thread dumps

System thread dumps provide a system view of a failing JVM (Java Virtual Machine) process. On Unixsystems, they usually appear as core files. On Window's systems they appear as drwtsn32.log files.

System dumps do not understand Java classes. Everything in a system dump is C library oriented. Thesystem dump information provided for JVM processes refers to Java's C libraries and not the referenceclass files.

System dumps should only be interrogated when a Java thread dump is unavailable. Pertinent informationcan be obtained from system dumps. However, mapping this information back into Java source code is verydifficult. The following sections explain how to interrogate the core and drwtsn32.log files. When theyare generated by the system, they need to be interrogated.

Unix platforms

Core files

Core files on Unix systems can be interrogated by dbx and gdb. Dbx is a tool that is part of theAIX install. On Sun, dbx can be installed for an additional expense. The gdb (GNU debugger)is freeware that can be downloaded.

Core file tips:

1. Ensure that the system core file size specification is unlimited.
2. Ensure that the file system containing the core file has enough space.

The following is a sample of how to use ulimit to verify and set the core dump size. If it is too small,a unusable core file will be generated.

Ulimit sample:

```
[pwh501]:root> ulimit -a
time(seconds) unlimited
file(blocks) unlimited
data(kbytes) unlimited
stack(kbytes) unlimited
memory(kbytes) unlimited
coredump(blocks) unlimited
nofiles(descriptors) 2000
```

The following commands will change the coredump (-c) and file (-f) to unlimited:

```
ulimit -f unlimited
ulimit -c unlimited
```

The following is an example of using the df command to verify that there isenough room in the file system for the core.The core file is placed in the ./bin directory. On AIX this is in the /usr filesystem.A core file can be 200MB.

Df sample:

```
[pwh501]:root> df
Filesystem 512-blocks Free %Used Iused %Iused Mounted on
/dev/hd4 131072 80416 39% 2480 8% /
/dev/hd2 8306688 2835096 66% 76320 8% /usr
/dev/hd9var 606208 55176 91% 390 1% /var
/dev/hd3 475136 459808 4% 32 1% /tmp
```

```
/dev/hd1 1310720 426120 68% 12453 8% /home

/dev/lv00 65536 47048 29% 96 2% /usr/lpp/netviewdm

/dev/lv01 606208 296504 52% 915 2% /db2

/dev/lv02 4014080 2806320 31% 3328 1% /Projects
```

Note: These samples were taken from the AIX 4.3.3 system.

DBX command

The purpose of the dbx command is to provide an environment to debug and run programs under the operatingsystem. The dbx command provides a symbolic debug program for C, C++, Pascal, andFortran programs, allowing you to carry out operations including:

- Examine object and core files
- Provide a controlled environment for running a program
- Set breakpoints at selected statements or run the program one line at a time
- Debug using symbolic variables and display them in their correct format

DBX syntax

```
dbx [ -a ProcessID ] [ -c CommandFile ] [ -d NestingDepth ] [ -I Directory ]

[-E DebugEnvironment ] [ -k ] [ -u ] [ -F ] [ -r ] [ -x ] [ ObjectFile

[ CoreFile ] ]
```

The ObjectFile parameter is an object (executable) file produced by a compiler. Use the -g (generate symbol table) flag when compiling your program to produce the information the dbx command needs.

Note: The -g flag of the cc command should be used when the object file iscompiled. If the -g flag is not used or if symbol references are removedfrom the xcoff file with the strip command, the symbolic capabilities ofthe dbx command are limited.

If the -c flag is not specified, the dbx command checks for a .dbxinit file inthe user's $HOME directory. It then checks for a .dbxinit file in the user'scurrent directory. If a .dbxinit file exists in the current directory, that file overrides the .dbxinitfile in the user's $HOME directory. If a .dbxinit file exists in the user's $HOME directory or currentdirectory, that file's subcommands run at the beginning of the debug session. Use an editor to createa .dbxinit file.

If ObjectFile is not specified, then dbx asks for the name of the object file to be examined. Thedefault is a.out. If the core file exists in the current directory or a core file parameter is specified,then dbx reports the location where the program failed. Variables, registers and memory held in thecore image may be examined until execution of ObjectFile begins. At that point the dbx debug programprompts for commands.

Note: The commands are referenced in the AIX Version 4.3 Commands Reference, Volume 2.

DBX tips

The common procedure of interrogating a core file is to change the directory to where the core fileresides. You can then issue the command with the binary executable file as the parameter. It isimportant that the binary executable is used. Usually the java command is a shell script that callsthe executable. If you enter the shell script, java, as the parameter a "cannot find" errormessage is returned.

The following commands show you how to find the binary executable and invoke thedbx command. It also shows an illegal instruction was executed (i.e. Invalid opcode):

```
[pwh501]:root> cd /usr/jdk_base

[pwh501]:root> find . -name java -print

./bin/aix/native_threads/java

./bin/java

[pwh501]:root> cd /usr/WebSphere/AppServer/bin

[pwh501]:root> ls -l core

-rw-r--r-- 1 root system 191495883 Aug 07 15:08 core

[pwh501]:root> dbx /usr/jdk_base/bin/aix/native_threads/java

Type 'help' for help.

Warning: The core file is truncated. You may need to increase the ulimitfor file and core dump, or free some space on
the file system.

Reading symbolic information ...Warning: no source compiled with -g [using memory image in core]

Illegal instruction (reserved addressing fault) in . at 0x0 ($t29)

0x00000000 00000001 Invalid opcode.
```

If you don't know where the java binary is located, the following command will display the true javaexecutable name of the core:

```
strings core | more
```

After you enter dbx, the where command provides a stack trace of where the error occurred. Thefollowing example shows a:

- Stack trace
- Output of the help command
- How to exit dbx

Stack trace

```
(dbx) where
warning: could not locate trace table from starting address 0x0
ExecuteJava(??, ??) at 0xd2f9913c
do_execute_java_method_vararg(??, ??, ??, ??, ??, ??, ??, ??) at 0xd2fabd30
execute_java_dynamic_method(0x20e355e0, 0x3002fdb0, 0xd3016aa4, 0xd3016aa8, 0x0, 0x0, 0x0, 0x0)
at 0xd2fabef4
ThreadRT0(0x3002fdb0) at 0xd300cd88
sysThread_shell(??) at 0xd2fb50a8
pthread._pthread_body(??) at 0xd010f358
```

Output of the help command

```
(dbx) help
Commands:
alias assign attribute call case catch
clear cleari condition cont delete detach
display(/) down dump edit file func
goto gotoi help ignore list listi
map move multproc mutex next nexti
print prompt quit registers rerun return
run rwlock screen search(/?) set sh
skip source status step stepi stop
stopi thread trace tracei unalias unset
up use whatis where whereis which

Topics:
startup execution breakpoints files data
machine environment threads expressions scope
set_variables usage
Type "help" for help on a command or topic.
```

How to exit dbx

```
(dbx) quit
[pwh501]:root>
```

Another useful purpose of the dbx command is to monitor a running process.The -a parameter allows the user to attach to a process. The `catch` and `run` commandscan be used to walk through the processing of the JVM process and see all signals that are caught. Useof the `help xxx` command will provide additional information on each of the abovecommands.

DBXTRACE.SH

There are shell scripts that call the dbx command and format the thread information from the core file.The name of the script is usually dbxtrace.sh. There is an AIX version and a Solaris version.

Here's a description on how to run the shell script:

```
[pwh501]:root> ./dbxtrace -a
Usage: Automate getting dbx trace information
For core files:
```

```
Usage: dbxtrace [executable] [core]

or : dbxtrace -c corefile

Example: dbxtrace /usr/jdk_base/bin/aix/native_threads/java core

(Please make sure you use the java executable and not the java script)

To attach to a running or hung process

Usage: dbxtrace -a PID

Example: dbxtrace -a 1234
```

The following information describes the beginning of the output when using dbxtrace on AIX:

```
[pwh501]:root> ./dbxtrace.sh | more
****************************************
* Failure of this script or dbx may *
* overwrite your existing core file. *
* It is recomended that you rename your *
* existing core file and use the -c flag *
* Do you wish to continue (y/n): ****************************************
Creating subcommand file....
Running dbx...
Type 'help' for help.
warning: The core file is truncated. You may need to increasethe ulimitfor file and coredump, or free some space on the
filesystem.
Reading symbolic information ...warning: no source compiled with -g
```

Note: The the user is prompted for (y/n). Therefore, if the user redirects the output toa file `[pwh501]:root> ./dbxtrace.sh > myfile 2>&1` a standalone "y"must be entered before the output is generated.

The output of the dbxtrace.sh provides information about each defined thread. The output has thefollowing sections:

- Error condition
- One line description of each thread
- Detail thread information
- Stack trace of each thread

Error condition:
Illegal instruction (reserved addressing fault) in . at 0x0 ($t29)
0x00000000 00000001 Invalid opcode.

One line description for each thread:
$t29 is the current thread
thread state-k wchan state-u k-tid mode held scope function
$t1 run blocked 37671 u no sys _pthread_ksleep
$t2 run blocked 38197 u no sys _pthread_ksleep
..
>$t29 run running 46443 k no sys

Detail thread information
thread state-k wchan state-u k-tid mode held scope function
>$t29 run running 46443 k no sys
general:
pthread addr = 0x20df04e0 size = 0x18c
vp addr = 0x20e376b4 size = 0x284
thread errno = 2
start pc = 0xf0545994
joinable = yes
pthread_t = 1c1d
scheduler:
kernel =
user = 1 (other)
event :
event = 0x0
cancel = enabled, deferred, not pending
stack storage:
base = 0x20df5738 size = 0x40000
limit = 0x20e35738
sp = 0x20e35040

Stack trace of each thread
thread state-k wchan state-u k-tid mode held scope function
*$t29 run running 46443 k no sys
warning: could not locate trace table from starting address 0x0
ExecuteJava(??, ??) at 0xd2f9913c

do_execute_java_method_vararg(??, ??, ??, ??, ??, ??, ??, ??) at 0xd2fabd30
execute_java_dynamic_method(0x20e355e0, 0x3002fdb0, 0xd3016aa4, 0xd3016aa8, 0x0, 0x0, 0x0, 0x0) at 0xd2fabef4
ThreadRT0(0x3002fdb0) at 0xd300cd88
sysThread_shell(??) at 0xd2fb50a8
pthread._pthread_body(??) at 0xd010f358

Windows platform

The drwtsn32.log files are similar to core files on Unix. On Windows 2000, these files arefound in the following directory C:\Documents and Settings\All Users\Documents\DrWatson.

After entering `drwtsn32 ?`, the "Dr. Watson for Windows 2000" box appears. The DrWatson logfile overview option will display a screen which explains the format of the drwtsn32.log files. Theoutput of the dbxtrace.sh provides information about each defined thread. The output has the samesection as a Unix platform:

- Error condition
- One line description of each thread
- Detail thread information
- Stack trace of each thread

Java thread dumps

Java thread dumps provide a Java view of a failing JVM process. Depending on the platform, Java dumpscan appear with different names and at different locations.

A Java dump provides information about the executing Java classes and allows the problem determinationprocess to reference the Java source code.

How to obtain a JAVA Thread Dump

There are two ways to obtain a Java thread dump:

- DrAdmin function
- kill -3 command

DrAdmin works on all platforms. On Unix, the kill -3command serves the same function and is easier to use. Therefore, DrAdmin is discussed in the Windowsplatform section and kill -3 is discussed in the Unix platforms section.

Unix platforms

Sometimes Java thread dumps will occur due to an error in the JVM. Atother times, the user might need to understand what is occurring within a JVM that is currently active.In either case, the Java thread dump is placed at the location described in locations table.Information on how to manually obtain a thread dump is available in the remainder of this section.

When a process hangs or is working hard (i.e. looping), it might be helpfulto understand what the individual threads of a JVM process are doing. Obtaining a stack trace of theindividual threads will provide this information. The kill -3 processid command provides this stack traceinformation. This command should not impact the running process.

Identifying process IDs

WebSphere supports four processes:

- Nanny - started with startupServer.sh
- Administrative server - started by the nanny process
- Administrative client - started with adminclient.sh
- Managed server - started by the administratiave server either automatically or manually via the administration client console.

These processes are usually started in the sequence that they are listed. Therefore, theirprocess IDs increase in value. The ps -ef | grep java command will display all the processes that areassociated with java.

The process IDs are listed under the second column in the command's output. The user can not use theadministration client interface if managed servers are automatically started by the administrative server.

Unfortunately, the ps -ef | grep java command does not always allow the user to identify the differentprocesses. The command string to start the processes can be very long and the length of the commandstring saved by the system may not be adequate for the ps -ef | grep java command.

On AIX, the complete command line is listed in the above ps -ef | grep command output.The user can also enter the following commands to focus on an individual process ID:

- ps -ef | grep Nanny
- ps -ef | grep AdminServer
- ps -ef | grep AdminClient
- ps -ef | grep ManagedServer

  There could be multiple managed servers running simultaneously. The managed server process ID(s) arealso displayed within the ./bin/tracefile with:

  Starting Server: "Default Server" (pid "116032")

  Default server is the name of the managed server. On the administration client window, the **General** tabinformation for the manged server also displays the process ID.

  As the root user, the kill -3 *xxxx* can now be entered where *xxxx* is the process ID ofthe WebSphere JVM in which you need to see a thread dump.

Location of thread dump

The location of the thread dump depends on the operating system.

Process	AIX 4.3.3	Sun OS 5.7	HP-UX B.11.0.0
Administrative server	./bin/javacore....txt	Appended to ./logs/tracefile	Appended to ./logs/tracefile
Managed server	./bin/javacore...txt	./Appended to stderr.file for managed server (Note 1)	Appended to stdout file for managed server (Note 1)
Administrative client	./bin/javacore...txt	Prompted at window used to enter adminclient.sh	Window used to enter adminclient.sh
Nanny	./bin/javacore...txt	Prompted at window used to enter startupServer.sh	Window used to enter startupServer.sh

Note 1: The stderr and stdout files are defined within the managed server configuration. Theconfiguration can be viewed with the administrative console. Click on the managed server (for exampledefault server). The standard output file and standard error file are defined within fields on the **General** Tab.

If the user starts a server in the background, the kill command may not dump the thread information.The workaround for this situation is to do the following:

startupServer.sh &

Ctrl-Z

fg

kill -3 xxxx

Windows platform

Create DrAdmin.bat

There is no DrAdmin.bat file shipped with WebSphere Application Server that will execute the DrAdmin function. One needs to be created.The following procedure is an example on how to create a DrAdmin.bat file. The procedure assumes that theWebSphere install is located at D:\WebSphere\AppServer. Follow these steps to create a DrAdmin.bat file:

1. Change the current directory to the directory containing adminserver.bat (cd D:\WebSphere\AppServer\bin\debug
2. Copy the adminserver.bat file to DrAdmin.bat in the same directory of adminserver.bat
3. Modify the newly created file, DrAdmin.bat, to execute the DrAdmin function. When changing the java commandline, the command needs to be on one continuous line. Modify the following line:

   %JAVA_HOME%\bin -DDER_DRIVER_PATH=%DER_DRIVER_PATH% -Xmx128m -Xminf0.15 -Xmaxf0.25 com.ibm.ejs.sm.server.AdminServer-bootFile %WAS_HOME%\bin\admin.config %restart% %1 %2 %3 %4

   with

   %JAVA_HOME%\bin\java com.ibm.ejs.sm.util.debug.DrAdmin %1 %2 %3 %4 %5 %6 %7 %8 %9

Find the port number of interest

The next step is to identify the port number for either the administratiave server or a managed server.The port number is different for the administratiave server and each of the managed server(s).The port number values are contained in the standard out files for each of these processes. Informationon how to find these files and the port number are described below.

After starting the administratiave server, you should obtain the DrAdmin port number within the.\logs\tracefile file inside the message:

```
DrAdmin available on port xxxx
```

After starting the managed server (for example, default server), you should obtain the DrAdmin port numberprovided in the standard output file for the managed server. The location of the file can be found withthe administrative console interface in the managed server configuration via the **General tab** to standard output field.The message within the file containing the port number is:

```
DrAdmin available on port xxxx
```

Execute DrAdmin

The DrAdmin.bat file can now be executed providing the port number obtained above. The format of thecommand to use is:

```
DrAdmin -serverPort xxxx -dumpThreads
```

where xxxx is the port number from the above message (without the comma).

Locate the thread dump

The administratiave server thread information is placed in .\logs\adminserver_stderr.log file.Because this file is not closed, it's length of 0 will not change. The application server thread informationis placed in the standard error file which can be found with the administrative console interface in theapplication server configuration via the **General** tab to standard error field. The thread information isdumped immediately into this file.

In order to view the thread information, copy the above files into a new file. Edit the new file with anHTML editor, which will display the thread information. Some editors (i.e. emacs and vi) will allow you toview the thread information directly from the .\logs\adminserver_stderr.log file or the standard error file.

How to interrupt a Java thread dump

A thread dump can be forced or can occur when a Java process error occurs. When a thread dumpis not forced, it usually means that an error within a Java process has occurred and it should to beinvestigated. A thread dump of a Java process needs to be forced when the process has a thread deadlockcondition. A **thread deadlock condition** is defined as:

```
Thread A currently owns Lock X.
Thread B owns Lock Y.
Thread A is waiting for the release of Lock Y in order to continue processing.
Thread B is waiting for the release of Lock X in order to continue processing.
```

Because of this stalemate condition, neither thread is able to complete its processing.

The referenced Java thread dump information is taken from a sample AIX dump. Java thread dumps on otherplatforms have similar information, but they may be formatted different.

Monitors

In order to have a thread safe application, the application may have to ensure that two threads don'texecute the same code simultaneously. This can be accomplished with

the use of a synchronized()statementor a synchronized modifier of a class method.

Each of the above will create a monitor/lock that will prevent other threads from executing the same code.It is important to understand that threads can be holding multiple monitors/locks while processing a request.Therefore two threads could find themselves in a deadlock condition defined by the following situation:

Thread A owns Lock X.

Thread B owns Lock Y.

Thread A is waiting for the release of Lock Y.

Thread B is waiting for the release of Lock X.

Because of this stalemate condition, neither thread is able to complete its processing.

Example of a deadlock condition

You can recognize a deadlock when looking within the native stack information. For example, when lookingat the native stack information of Thread A you can easily recognize that it is blocked by a monitor/lockheld by Thread B. This information does not appear in the native stack information of the Thread B.Thread B is currently deleting a connection, (deleteConn), from the ConnectionTable.The deleteConn()is a synchronized method which causes a monitor/lock to occur for the ConnectionTable class. There is only one **ConnectionTable** instance. Therefore the monitor/lock held by **Thread-9** is preventing the **JavaIDL Listener** from completing.

The above diagnosis requires an understanding of the involved source code (i.e. which methods aresynchronized). However, the Java thread dump does provide the pointers to do this additional investigation.

A summary of the object monitors will provide additional information that identifies Thread A as blockedThread B with:
```
com.ibm.CORBA.iiop.IIOPConnection@4fe89740: owner: "Thread B""Thread A" (0x36951ba8) blocked
```

Unfortunately, this information does not appear in the summary for the monitor being held by Thread A.

Stack traces

Stack traces represent the current call path of a thread. Call path information explains whatfunctional calls were made to get to the thread's current location.

System dump stack trace

Note that the sysAcceptFD() call is the last function called on the stack. Itis a system call that was invoked by java_net_PlainSocketImpl_socketAccept() call. The call indicatesthat a Java thread did an accept operation on a socket. Question marks appear as parameters.This is because the Java process was not run in debug mode. For Java 1.1 installations(i.e. before WebSphere 3.5), debug mode is started by using the java_g command. For Java 1.2 installations(i.e. WebSphere 3.5), the -Xdebug options should be used with the Java command. For either type ofinstallation, the administration console screen for a managed server configuration has a Debug tab.Debug mode can be set within this tab. As stated above, no class file information appears in the stacktrace. Only the functions with C libraries are referenced.

Java dump stack trace

The reader is able to follow the sequence of calls fromthe run() method through the read() method of the SocketInputSteam class. The package names of theclasses are also present. The "Compiled Code" characters appears as parameters in the callbecause the Java dump occurred for a JVM that was not running in debug mode. When running in debug mode,the line number of the call within the source replaces the "Compiled Code" characters. ForJava 1.1 installations (i.e. before WebSphere 3.5), debug mode is started by using the java_g command.For Java 1.2 installations (i.e. WebSphere 3.5), the -Xdebug options should be used with the java command.For either type of installation, the administration console screen for a managed server configurationhas a Debug tab. Debug mode can be set with this tab. Another way of obtaining the source line numberis to turn off the JIT (Just In Time) compiler. This can be done by starting the JVM with the-Djava.compiler=NONE parameter. This parameter can also be placed on the managed server command line.The command line information can be displayed with the administration console interface on the **General** tabof the managed server configuration.

WebSphere Application Server thread information

Object Request Broker information

During startup of the different WebSphere Application Server processes, the processes are initialized andplaced in a state to accept additional network activity. One of the first steps in initializing aprocess is to create an ORB instance. This step will create threads that will be used to complete theinitialization step and later accept network activity to be processed.

These activities are described within each of the two diagrams of the next two sections. The diagramsdescribe:

- Administratiave server startup and immediate administrative server takedown
- Managed server startup with servlet traffic

The administratiave server has two ORBs defined within it. For each ORB there is at least one ORBserver listener thread that continually waits for input on a port. When input is received, it is dispatchedto a ORB server reader thread so the ORB server listener thread can again wait for input on the port. TheORB server reader thread again dispatches the request to a third thread that completes the work activity.The reply to the work activity is sent from the third thread to a ORB client reader thread that receivesreplies from the ORB reader thread. An ORB request has four steps/threads involved:

1. ORB Server Listener thread receives input on port X.
2. ORB Server Reader thread is given a request.
3. Pooled/instantiated thread handles the request and sends a reply.
4. ORB Client Reader thread handles the reply.

The managed server has one ORB defined within it. There are two ORB server listener threads andmultiple ORB client threads. The servlet traffic does not use the ORB for communications. It is done withthe plug-in interface. This interface supports a pool of worker threads (Worker#_) that complete theHTTP requests.

The following port numbers are preset:

- 9000 is used for obtaining naming services (i.e. data source names, EJB names)
- 900 is used by the administratiave server to listen for administrative client requests

Other port numbers are randomly chosen for ORB communications.

Thread names

ORB threads

The ORB instance creates reader and listener threads. The names of these threads get changed afterthey begin processing (i.e. during run() method processing). The name is constructed with the followingparameters separated by a colon (:):

- ORB information
  - P = unique for this process and algorithmically constructed from a time stamp
  - O = number of ORB's within this process
- Thread type
  - StandardRT = identifies which reader thread is within the ORB
  - CT = client thread
  - LT = listener thread
- Connection values
  - LocalPort = Local port that thread is dealing with
  - RemoteHost = Hostname for ORB server reader thread, or IP address for ORB client reader thread
  - RemotePort = Port number on the remote host for the connection

Worker#__ (SERVLET ENGINE THREADS)

These thread names begin with Worker# and process HTTP requests.

Thread-x

These thread names are the default thread name for Windows 2000 and AIX. Because no thread name isprovided this name is used. X is incremented as each new thread is created.

Pooled ORB request dispatch WorkerThread

These threads are created by the main thread (i.e. P=479481:O=0:CT)and handle the request/repliesthat are sent across IIOP connections.

Web server plug-in configuration thread

Thread used for setting the Web server configuration.

Alarm manager

This thread manages the creation of alarm thread x's.

Alarm thread 1

The alarm thread 1 reclaims unused connections.

BackgroundLruEvictionStrategy

This thread sweeps a cache, reclaiming the least recently used objects.

Refresh

This thread insures that any changes to a model get propagated to clones.

Thread stack traces

When a thread is created, the start() method is used to invoke the run() method. The start() method isexecuted on one thread and the run() method is executed on the newly created thread. Dependingon when the stack trace is obtained, an activity (i.e. piece of work) could have different stack traces.Therefore, thread names have two base method calls. The following text describes these base method callsfor the common thread names used for both the administratiave server and the managed server. Two stacktrace examples of base method calls are also provided:

Base method calls

1. Main or P=xx:O=0:CT
   - run ---> com.ibm.ejs.sm.server.AdminServer.main()

2. ORB server listener thread (JavaIDL Listener or P=xx:O=0:LT=0:port=9000)
   - start ---> com.ibm.ejs.sm.server.AdminServer.main()
   - run ---> com.ibm.CORBA.iiop.ListenerThread.run()

3. ORB server reader thread (JavaIDL Reader for hostname:port# or P=xx:O=0:StandardRT=0:LocalPort=port#:RemoteHost=hostname:RemotePort=port#:)
    - start ---> com.ibm.CORBA.iiop.ListenerThread.run()
    - run ---> com.ibm.CORBA.iiop.StandardReaderThread.run()
4. ORB client reader thread (JavaIDL Reader for ipaddr:port# or P=xx:O=1:StandardRT=1:LocalPort=port#:RemoteHost=ipaddr:RemotePort=port#:)
    - start ---> com.ibm.ejs.sm.server.AdminServer.main()
    - run ---> com.ibm.CORBA.iiop.StandardReaderThread.run()
5. Pooled ORB request dispatch WorkerThread
    - start ---> com.ibm.CORBA.iiop.StandardReaderThread.run()
    - run ---> com.ibm.ejs.oa.pool.ThreadPool$PooledThread.run()
6. Worker#__
    - start ---> com.ibm.ejs.sm.server.AdminServer.main()
    - run ---> com.ibm.servlet.engine.oselistener.outofproc.OutOfProcThread$CtlRunnable.run() java.lang.Thread.run()
7. Web server plug-in configuration thread
    - start ---> com.ibm.ejs.sm.server.AdminServer.main()
    - run ---> com.ibm.servlet.engine.oselistener.outofproc.OutOfProcThread$CtlRunnable.run() java.lang.Thread.run()
8. Alarm manager
    - start ---> com.ibm.ejs.sm.server.AdminServer.main() <--AdminServer
    - com.ibm.ejs.oa.pool.ThreadPool$PooledThread.run() <--AppServer
    - run ---> com.ibm.ejs.util.am.AlarmManagerThread.run() java.lang.Thread.run()
9. Alarm thread 1
    - start ---> com.ibm.ejs.util.am.AlarmManagerThread.run() java.lang.Thread.run() <--AdminServer
    - com.ibm.ejs.oa.pool.ThreadPool$PooledThread.run() <--AppServer
    - run ---> com.ibm.ejs.oa.pool.ThreadPool$PooledThread.run() <--AdminServer
    - com.ibm.ejs.util.am.AlarmThread.run() <--AppServer
10. BackgroundLruEvictionStrategy
    - start ---> com.ibm.ejs.sm.server.AdminServer.main()
    - run ---> com.ibm.ejs.util.cache.BackgroundLruEvictionStrategy.run()
11. RefreshThread
    - start ---> com.ibm.ejs.sm.server.AdminServer.main()
    - run ---> com.ibm.ejs.wlm.server.config.ServerGroupRefresh$RefreshThread.run()

Examples

Thread dump of a standard reader thread:

```
"P=863240:O=1:StandardRT=16:LocalPort=10502:RemoteHost=gofast:RemotePort=2619:"
(TID:0x11ccef0, sys_thread_t:0xcdd81d0, state:R, native ID:0x128) prio=5
>at java.net.SocketInputStream.socketRead(Native Method)
at java.net.SocketInputStream.read(SocketInputStream.java(Compiled Code))
at com.ibm.rmi.iiop.Message.readFully(Message.java(Compiled Code))
at com.ibm.rmi.iiop.Message.createFromStream(Message.java:173)
at com.ibm.CORBA.iiop.IIOPConnection.createInputStream(Unknown Source)
at com.ibm.CORBA.iiop.StandardReaderThread.run(Unknown Source)
```

The base method, com.ibm.CORBA.iiop.StandardReaderThread.run(), is identified as the run base method forJavaIDL Reader for hostname:port# threads. Also, the thread is waiting for input because it isin the java.net.SocketInputStream.socketRead() method.

Thread dump of a worker thread:

```
"Worker#49" (TID:0x10793660, sys_thread_t:0xab25b0, state:R, native ID:0x19a) prio=5
at com.ibm.servlet.engine.oselistener.outofproc.NativeServerQueueImp.nativeGetSeviceMessageId()
at com.ibm.servlet.engine.oselistener.outofproc.NativeServerQueueImp.getSeviceMessageId()
at
com.ibm.servlet.engine.oselistener.serverqueue.SQWrapperEventSource$SelectRunnable.getNewConnectionFromQueue()
at com.ibm.servlet.engine.oselistener.serverqueue.SQWrapperEventSource$SelectRunnable.run()
at com.ibm.servlet.engine.oselistener.outofproc.OutOfProcThread$CtlRunnable.run()
at java.lang.Thread.run()
```

The base method, java.lang.Thread.run(), is identified as the run base method for Worker#__ threads.Also, the thread is waiting for input from the Web server plug-in (native code) because it is in thecom.ibm.servlet.engine.oselistener.outofproc.NativeServerQueueImp.nativeGetSeviceMessageId() method.

Administrative Server Startup with Immediate Takedown Diagram

The following diagram has highlighted request flows that start with a SendReqXXX where XXX is the portnumber of the send request. The steps in the flow changes between different threads. The sequence of thesteps are identified with, for example, 1A,1B, 1C and 1D. It also shows how the port that the request issent to determines which thread the proccessing has completed.

Diagram Legend

In each diagram, every continuous line (-------) is a thread. The name of the thread always appearsbetween (...). The letters in the diagram have the following meanings:

C = Thread name changed to (.....)
S = Start method called on this thread

R = Run method called on this thread
W = Thread is in wait state waiting for notify
WM = Thread is waiting for message from plugin (Worker# threads only)
SendReq____ = Request sent to port number (____)
SendReply___ = Reply sent to port number (____)

For example:

C(P=479481:O=0:CT) = thread name is changed to P=479481:O=0:CT
R = thread is placed in a running state

Diagram

ORB 0 Threads(i.e. O=0)

main

|

|

C(P=479481:O=0:CT)

|

|S(JavaIDL Listener) R C(P=479481:O=0:LT=0:port=9000)

|----------> | |

| S(JavaIDL Reader for rbostick:1294)

| |

| |

| R

| |

| |

| C(P=479481:O=0:StandardRT=0:LocalPort=9000:

| RemoteHost=rbostick:RemotePort=1294:)

| |

| **1B**

| |S(Thread-1) R

SendReq9000(**1A**) |-**1C**---------->

| |

| **2B**

| |S(Thread-2) R

SednReq9000(**2A**) |-**2C**---------->

| |

| **5B**

| |

| |

| |

| V

|

|S(JavaIDL Reader for 9.27.63.245:9000) R C(P=479481:O=1:StandardRT=1:LocalPort=1294:

| RemoteHost=9.27.63.245:RemotePort=9000:)

|------**1D**--**2D**--**5D**----------> |

|

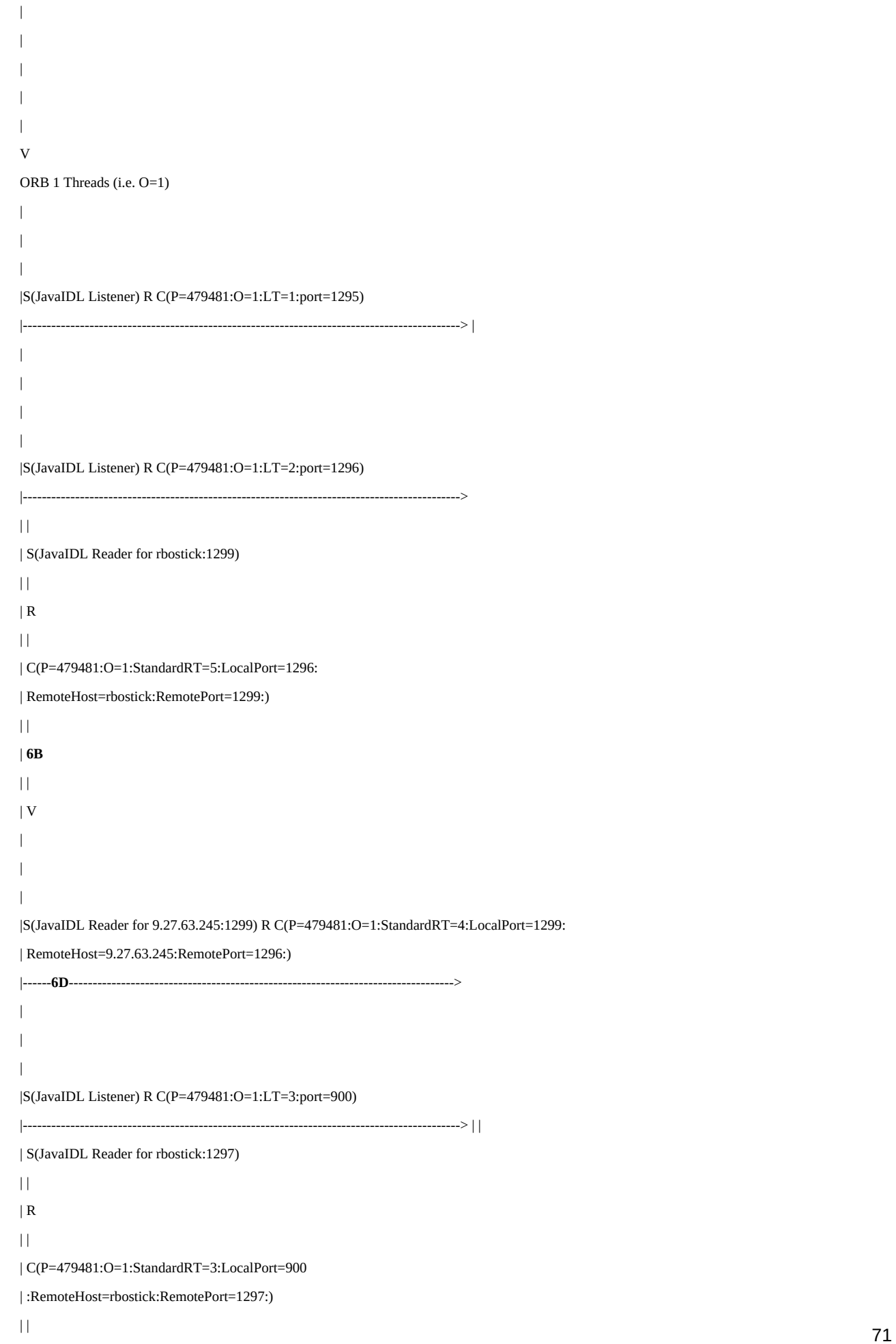
|

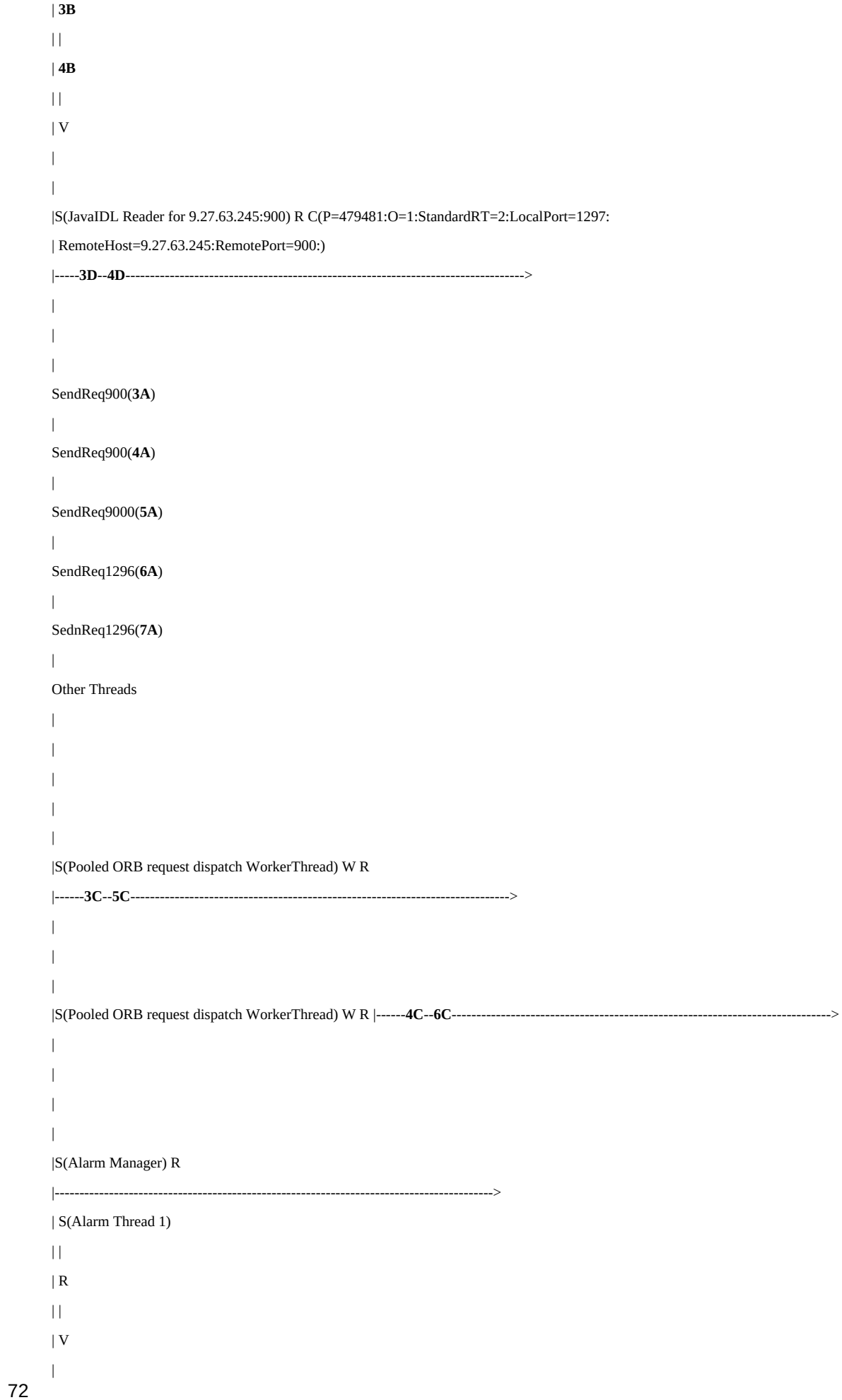
|

|

|

|


|
|
|
|
|
V
ORB 1 Threads (i.e. O=1)
|
|
|
|S(JavaIDL Listener) R C(P=479481:O=1:LT=1:port=1295)
|----------> |
|
|
|
|S(JavaIDL Listener) R C(P=479481:O=1:LT=2:port=1296)
|---------->
| |
| S(JavaIDL Reader for rbostick:1299)
| |
| R
| |
| C(P=479481:O=1:StandardRT=5:LocalPort=1296:
| RemoteHost=rbostick:RemotePort=1299:)
| |
| 6B
| |
| V
|
|
|
|S(JavaIDL Reader for 9.27.63.245:1299) R C(P=479481:O=1:StandardRT=4:LocalPort=1299:
| RemoteHost=9.27.63.245:RemotePort=1296:)
|------6D---------->
|
|
|
|S(JavaIDL Listener) R C(P=479481:O=1:LT=3:port=900)
|----------> | |
| S(JavaIDL Reader for rbostick:1297)
| |
| R
| |
| C(P=479481:O=1:StandardRT=3:LocalPort=900
| :RemoteHost=rbostick:RemotePort=1297:)
| |


| 3B
| |
| 4B
| |
| V
|
|
|S(JavaIDL Reader for 9.27.63.245:900) R C(P=479481:O=1:StandardRT=2:LocalPort=1297:
| RemoteHost=9.27.63.245:RemotePort=900:)
|-----3D--4D---------->
|
|
|
SendReq900(3A)
|
SendReq900(4A)
|
SendReq9000(5A)
|
SendReq1296(6A)
|
SednReq1296(7A)
|
Other Threads
|
|
|
|
|
|S(Pooled ORB request dispatch WorkerThread) W R
|------3C--5C---------->
|
|
|
|S(Pooled ORB request dispatch WorkerThread) W R |------4C--6C---------->
|
|
|
|
|S(Alarm Manager) R
|---------->
| S(Alarm Thread 1)
| |
| R
| |
| V
|

|S(Thread-3) R

|-->

|

|

|S(Thread-4) R

|-->

| | | | | |

| S(Thread-8) S(Thread-9) S(Thread-10) S(Thread-11) S(Thread-12) | | | | | |

| R R R R R

| | | | | |

| V V V V V

|

|

|S(Worker#0) R S(Worker#0) R

|-->

|

|

|S(WebServer-Plugin-Cfg-Thread) R

|-->

|

|S(BackgroundLruEvictionStrategy) R

|-->

|

|S(RefreshThread) R

|-->

V

Thread-1(2) (worker threads)

- start ---> com.ibm.CORBA.iiop.StandardReaderThread.run()run ---> com.ibm.CORBA.iiop.WorkerThread.run()

Thread-3 (transaction timeout)

- start ---> com.ibm.ejs.sm.server.AdminServer.main()
- run ---> com.ibm.ejs.jts.tran.JavaClock.run()

Thread-4 (used for AdminServer takedown)

- start ---> com.ibm.ejs.sm.server.AdminServer.main()
- run ---> com.ibm.ejs.sm.server.ManagedServer$DiagonisticThread.run()

Thread-8,9,10,11,12 (threads for takedown process)

>

- start ---> com.ibm.ejs.sm.server.ManagedServer$DiagonisticThread.run()
- run ---> com.ibm.ejs.sm.util.task.AsyncTaskEngine$WorkerThread.run()

Note: Thread-x are default names of threads. The above numbers may be different depending onthe system that the administratiave server runs on.

Managed Server Startup with Servlet Traffic Diagram

The Worker#_threads are the threads on which servlet requests are processed. The threadsstart during the managed server startup and wait on input from the Web server plug-in interface.

main

|

|

C(P=905990:O=0:CT)

|

|

|S(Thread-0) R

|-->

|

|

|S(Pooled ORB request dispatch WorkerThread) W R

|-->

| | | | | | | |

| S | | S(**Worker#0**)S(**Worker#1**).....S(**Worker#24**) S(Thread-6) (BackgroundLruEvictionStrategy) | | | |

| | | R WM WM WM R

| R | | | | |

| | S(AlarmManager) | S(**Worker#0**) | **Servlet** |

| | | | | **Request** |

| | | S(pluginRegenScheduler) | | | |

| | R | WM | WM |

| | | V V V V V

| | |

| | |

| V |S(AlarmThread1) R

| |---------------------------->

| V

|

|

|S(Pooled ORB request dispatch WorkerThread) W R W |-->

|

|S(Thread-1) R

|-->

|

|

|

|S(Thread-3) R

|-->

|

|

|

|

ORB 0 Threads (i.e. O=0)

|

|S(JavaIDL Reader for 9.27.63.129:9000) R C(P=905990:O=0:StandardRT=0:LocalPort=1480:

| RemoteHost=9.27.63.129:RemotePort=9000:

|-->

|

|

|

|S(JavaIDL Listener) R C(P=905990:O=0:LT=0:port=1481)

|-->

|

|

|S(JavaIDL Reader for 9.27.63.129:1434) R C(P=905990:O=0:StandardRT=1:LocalPort=1482:

| RemoteHost=9.27.63.129:RemotePort=1434:)

|-->

|

|

|

|

|S(JavaIDL Reader for 9.27.63.129:900) R C(P=905990:O=0:StandardRT=2:LocalPort=1483:

| RemoteHost=9.27.63.129:RemotePort=900:)

|-->

|

|

|

|S(JavaIDL Reader for 9.27.63.129:1433) R C(P=905990:O=0:StandardRT=3:LocalPort=1484:

| RemoteHost=9.27.63.129:RemotePort=1433:)

|-->

|

|

|

|

|S(JavaIDL Listener) R C(P=905990:O=0:LT=1:port=1485)

|-->

| |

| S(JavaIDL Reader for rbostick:1487)

| |

| R

| |

| C(P=905990:O=0:StandardRT=4:LocalPort=1485:

| RemoteHost=rbostick:RemotePort=1487:)

| |

| |

| V

V

Thread-0 (transaction timeout)

- start ---> com.ibm.ejs.sm.server.ManagedServer.main()
- run ---> com.ibm.ejs.jts.tran.JavaClock.run()

Thread-1 (Used for logging messages)

- start ---> com.ibm.ejs.sm.server.ManagedServer.main()
- run ---> com.ibm.ejs.sm.server.SeriousEventListener$DeliveryThread.run()

Thread-3

>

- start ---> com.ibm.ejs.sm.server.ManagedServer.main()
- run ---> com.ibm.ejs.sm.server.ManagedServer.main()

Thread-6 (AdminServer ping)

- start ---> com.ibm.ejs.oa.pool.ThreadPool$PooledThread.run()
- run ---> com.ibm.ejs.sm.server.ManagedServer$PingThread.run()

Note: Thread-x are default names of threads. The above numbers may be different. depending on the system the managed server runs on.

Summary

In multi-processing and multi-thread environments, problem determination can require analysis ofactively running threads. This thread information can be obtained with system thread dumps and Java threaddumps. When doing problem determination in a WebSphere Application Server environment, Java thread dumpsprovide much more information and are recommended. However, sometimes system thread dumps are the onlyinformation obtained and should be interrogated.

When dealing with thread deadlock problems, Java thread dumps can be forced using kill -3 on Unixplatforms and DrAdmin on all platforms.

The output of these commands provides thread information necessary to diagnose the problem.

8.10: Applying e-fixes

E-fixes are individual fixes for critical problems. They have been individually tested,but not integration tested and should only be applied if you have a critical problemwithout a valid workaround. They may be applied to both versions of WebSphere, exceptwhere specifically noted. All e-fixes are rolled into the next scheduled FixPack.Each fix has a readme file with installation instructions.

To learn about the fixes made available since the last FixPack, see the FixPacks and E-fixes website.

8.11: Resource reference

Use these links to learn about other performance tools and techniques.

- 9.1: Tuning the Product

8.13: Problem determination hints and tips

When you encounter an error or problem with WebSphere Application Server, you can followthe Hints and Tipsto help you quickly gather relevant data to diagnose theproblem.

8.14: How to report a problem to IBM

Use the information in this section to help you report a problem to IBM.

Before reporting problems to IBM, please review the known problems in the Release Notes,Hints and Tips, FAQ's, and other resources on the support website. If you find that the problem is not a known defect, then report the problem to IBM.

There are a variety of ways to report your problem to IBM:

- Phone
- Fax
- Internet

If you need assistance with problems, you are required to purchase technical support. You can select the exact mixof services to fit your specific business needs. IBM Software Support is delivered in a consistent manner for all IBMsoftware products based upon the way in which a product is charged (one time charge or monthly license charge basis).

You can report suspected defects via fax, mail or electronically until the product's service expiration date. Thisfree service is called Warranty/Defect Support. For information on reporting suspected defects, call 1-800-237-5511 in theUnited States and Puerto Rico. In Canada, call 1-800-465-9600. Telephone numbers for countries outside North Americaare also available. The service expiration date is defined in your License Information booklet under Program Services.

What to provide when reporting problems

You will need the following information available when reporting a problem to IBM:

- The product name and version number
- The kind of hardware and software you are using
- What happened and what you were doing when the problem occurred
- Whether you tried to solve the problem and how
- The exact wording of any messages displayed

After you have reported a problem to IBM support using any of the methods above, especially by phone, you might wantto provide relevant logs, traces or files. You can also send an ASCII text description of the problem in your own words.Send logs and text files together in a zip file for ease of transfer.

Follow these steps to send files to IBM:

1. Note the problem record number assigned to you by IBM support.
2. FTP testcase.software.ibm.com
3. Login: anonymous
4. Password: [your email id]
5. Change directory: cd /ps/toibm/internet
6. Make a directory: mkdir pmrnumber [use your problem number, for example, pmr89401]
7. Put [filename]
8. Call IBM Support back and ask that it be noted in your problem record that files are available on the testcase ftp server. Give the path to the files. Files will remain available on the testcase ftp server for 72 hours and will then be deleted.

Technical support by phone

If you are a licensed customer in the U.S. or Puerto Rico who has a support contract and youneed support, please call IBM Support at 1-800-237-5511. In Canada, call 1-800-IBM-SERV (1-800-426-7378). Telephonenumbers for countries outside North America are also available.

If you are a licensed customer and wish to purchase support, you may contact IBM or yourIBM authorized business partner.

If you have an IBM customer number, call 1-888-426-4343 Monday - Friday 8:00 a.m. to 7:00 p.m. Eastern Standard Time.
In Canada, call 1-800-465-9600 Monday - Friday 8:00 a.m. to 5:00 p.m. Central Standard Time.

If you do not have an IBM customer number, call 1-800-237-5511 Monday - Friday 8:00 a.m. to 5:00 p.m. Central Standard Time.
In Canada, call 1-800-465-9600 Monday - Friday 8:00 a.m. to 5:00 p.m. Central Standard Time.

Technical support by fax

Contact us via the Faxback System: 1-800-426-4329.
Telephone numbers for countries outside North America are also available.

Technical support on the Internet

Online help is available through the IBM Support Line.Support Line is the service offering through which IBM delivers electronic support for installation, usage, andcode-related questions. Electronic support is also available through Passport Advantage'sonline incident report page. Solution developers can alsoreceive online help through the PartnerWorldfor Developers.

Information on IBM SupportLine and IBM Services is available on the Internet at the URL listed above. For IBMLotus Passport Advantage customers, support information is also available at this Internet site.

Note: Information may not apply to all products. Support information is subject to change without notice.