WebSphere software

IBM

Redbooks Paper

Carla Sadtler
David Titzler

WebSphere Application Server V6: Diagnostic Data

This paper contains information about the diagnostic data that is available in WebSphere Application Server V6. It contains information about the location of the data, how it is collected, and configuration options.

It includes information about the following:

- JVM logs (SystemOut and SystemErr)
- Tracing
- Collector tool
- First Failure Data Capture (FFDC)
- Process (native) logs
- Service log (activity.log)
- Installation logs
- IBM HTTP Server and plug-in logs and traces
- System management logs
- WebSphere Rapid Deployment logs

JVM logs (SystemOut and SystemErr)

SystemOut and SystemErr logs are created for every WebSphere® Application Server process (application server, cluster member, node agent, and deployment manager). These logs are known as JVM™ logs. The System.Out and System.Err streams for each JVM are redirected to the SystemOut and SystemErr logs. WebSphere Application Server writes to these logs. Your applications can also write to them by using the print(), println(), and printStackTrace() methods.

You can find the logs in the following directory:

<WAS_install_root>/profiles/*<profile>*/logs/*<process>*

To configure the properties of these logs from the administrative console, select **Troubleshooting** → **Logs and Trace**. Select the process whose logs you want to configure, and then click **JVM Logs**. The General Properties window opens, as shown in Figure 1.

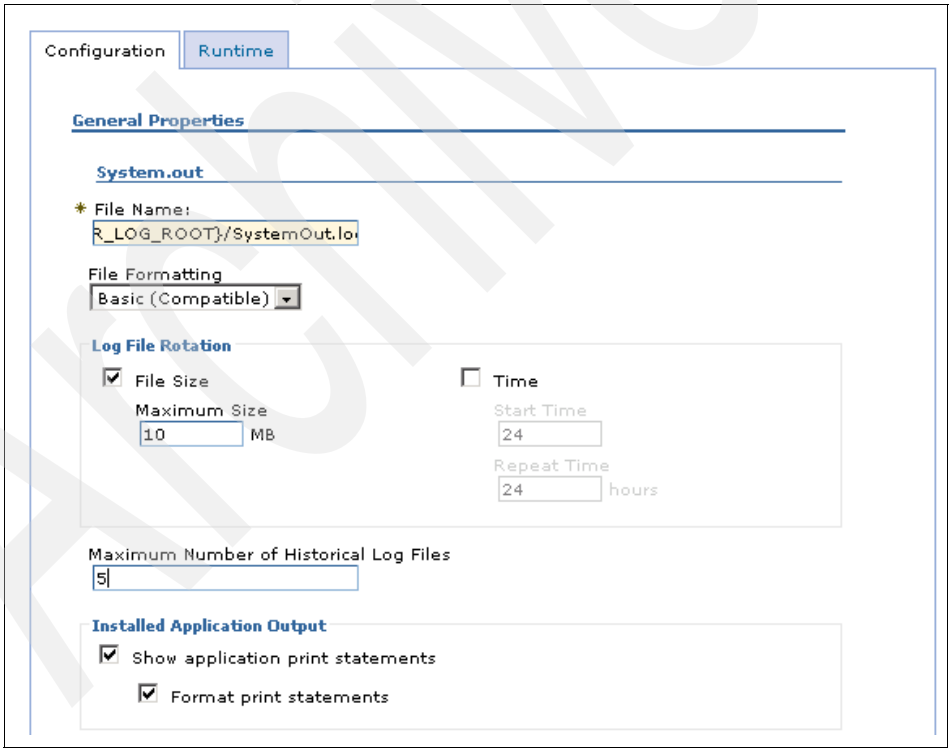

Figure 1 Changing the log file rotation properties

In the Configuration tab, you can edit the following:

- The file name (and the directory) of the SystemOut and SystemErr logs
- The file formatting

  We recommend that you leave this at the default value of Basic to make the logs easier to read.
- Log file rotation

  The SystemOut and SystemErr logs are self-managing. They write to the specified file until either the maximum file size or a certain time is reached. When that happens, the current log file is renamed as the current file name plus the current time stamp. Then a new SystemOut or SystemErr file is created for further logging. The older log files are called historical log files.

  For example, after this occurs you might have the following SystemOut files in the *<WAS_install_root>*/profiles/*<profile>*/logs/*<process>* directory:

  – SystemOut.log, the current log file
  – SystemOut_05.06.07_10.28.48.log, the historical log file

  Depending on your needs, you can choose to have the log files rotate (roll over) when they reach a specified size, a certain time interval, or both. If you choose a time, we recommend that you specify 24 hours as the Repeat Time. You can also set the Start Time to specify the time at which the logs rotate.

  If you specify a file size, we recommend that you increase the Maximum Size above its default of 1 MB. You will want to coordinate the value of Maximum Size with the Maximum Number of Historical Log Files, based on the available disk space on your system. With either method, make sure that the amount of log data that is saved is enough so that the relevant log data is there when you identify that a problem has occurred.
- Maximum Number of Historical Log Files

  The value that is entered here is the number of historical log files that are kept. If the value is reached and another historical log file needs to be created, the oldest one is removed from your system.
- Installed Application Output

  These properties affect how `print` and `println` statements from your applications are output. There are two options:

  – Show application print statements. This is enabled by default. If you deselect it, application `print` and `println` statements are not logged to the SystemOut and SystemErr log files.

  – Format print statements. This is also enabled by default. You can deselect it if you do not want your application `print` and `println` statements to be

formatted similar to WebSphere Application Server messages in the log files.

All of these properties can be changed for both the SystemOut and the SystemErr logs. You can choose to use the same properties for both logs, which we recommend, or use different properties for them.

You can view the SystemOut and SystemErr files on the file system with a text editor or you can view them within the administrative console. It might be useful to view them in the administrative console if you need to view them from a remote system. To view them in the administrative console, select the Runtime tab and then click **View** (next to the SystemOut or SystemErr file name).

The entries in the output of the SystemOut.log are in the following format:

```
[7/12/05 14:46:00:264 EDT] 0000001a ApplicationMg A   WSVR0221I: Application
started: adminconsole
```

Each entry can be deciphered as follows:

- Time stamp

  In the example, the time stamp is `[7/12/05 14:46:00:264 EDT]`.

  The time stamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example MM/DD/YY), 24-hour time with millisecond precision, and a time zone.

- Thread ID

  In the example, the thread ID is `0000001a`.

  The thread ID is an eight-character hexadecimal value that is generated from the hash code of the thread that issued the message.

- Short name

  In the example, the short name is `ApplicationMg`.

  The short name is the abbreviated name of the component that issued the message. This name is typically the class name of a WebSphere Application Server component and would be some other identifier for applications.

- Event type

  In the example, the event type is `A`.

  The event type is a one character field that indicates the type of the message. The possible values are:

  - F - fatal message
  - E - error message
  - W - warning message
  - A - audit message
  - I - informational message
  - C - configuration message
  - D - detail message
  - O - message that was written directly to System.out by an application or internal components
  - R - message that was written directly to System.err by the user application or internal components
  - Z - a placeholder to indicate that the type was not recognized

- Message Identifier

  In the example, the message identifier is `WSVR0221I`.

  The message identifier is a string that is nine characters in length and is in the form `CCCC1234X`. The first four characters (`CCCC`) indicate the WebSphere Application Server component that issued the message. The next four characters (`1234`) indicate the specific message that the component is issuing. The last character (`X`) indicates the severity of the message. Its value is either `I` (informational), `W` (warning), or `E` (error).

  You can find descriptions of all WebSphere Application Server message identifiers in the WebSphere Information Center item *Troubleshooter reference: Messages* at:

  `http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/welc_ref_trb_msg.html`

- Message

  In the example, the message is `Application started: adminconsole`.

  The message is the data that is logged to the SystemOut.log by the component. It is meant to provide useful output for informational purposes, debugging, and troubleshooting.

Tracing

Trace logs can also be configured in a manner similar to the JVM logs. Traces must be explicitly enabled. They are disabled by default. Trace output gives very detailed information about the execution of WebSphere Application Server code. It provides time stamps, details about which WebSphere methods were called, and special diagnostic data that is included to make troubleshooting easier.

To configure the properties of the traces from the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the process whose trace logs you want to configure.
3. Click **Diagnostic Trace**. The General Properties window opens, as shown in Figure 2.

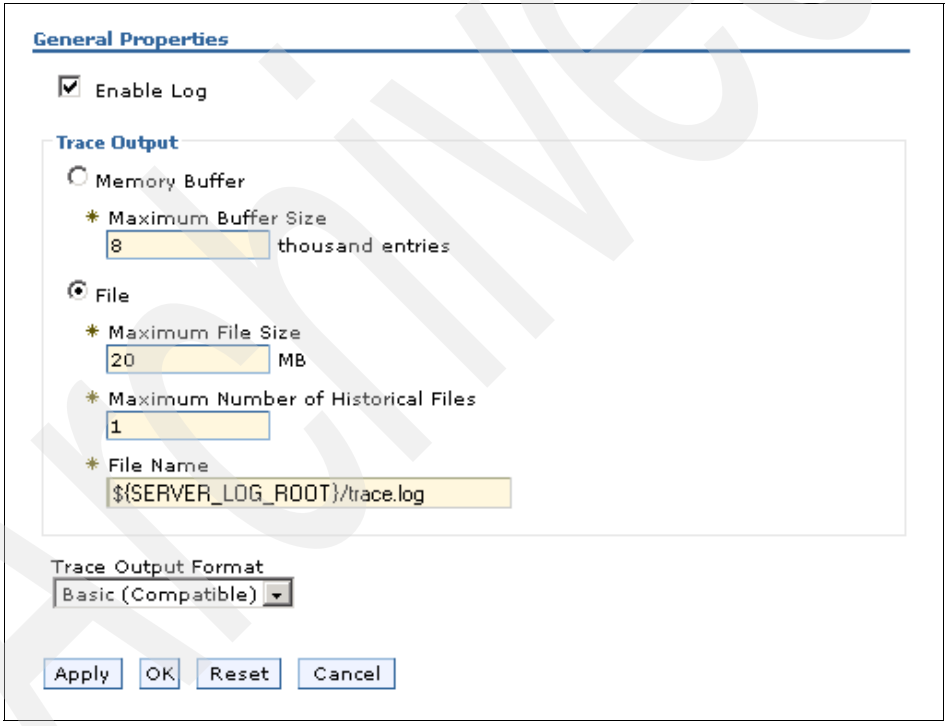

Figure 2 Changing the trace properties

The configuration of the trace file properties is very similar to the configuration of the JVM logs. You want to ensure that Enable Log remains selected. For Trace Output, we recommend that you always select File instead of Memory Buffer so that the trace logs are easier to manage.

Trace files cannot be rolled over based on time. You must specify a Maximum File Size in conjunction with the Maximum Number of Historical Files. You should set these values appropriately, depending on how long it might take to reproduce the problem with trace enabled and how much disk space is available. With these properties set, the trace files roll over in the same manner as the JVM logs. You can also specify a File Name and specify a directory for the trace files.

As with the JVM logs, we strongly recommend selecting Basic (the default value) for the Trace Output Format. This makes the trace easier to read, and it is the preferred format of the WebSphere Application Server support team.

When you view the trace properties in the administrative console, you notice that there are two tabs, a Configuration tab and a Runtime tab. You can enable tracing on either tab. The difference is that when you use the Configuration tab, you must restart the WebSphere Application Server process before the tracing begins. When you use the Runtime tab, tracing begins as soon as you click **OK** or **Apply**. In many production environments, it is preferable to enable trace using the Runtime tab so that you do not have to restart the WebSphere process.

After the trace properties are configured, you must decide which WebSphere Application Server components to trace. To do this in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the process whose trace logs you want to configure
3. Click **Change Log Level Details**.

   As with the trace properties, the log level details can be set on the Configuration tab or the Runtime tab.

   In the Change Log Detail Levels screen (Figure 3 on page 8), you can select which *components* and *groups* to trace.

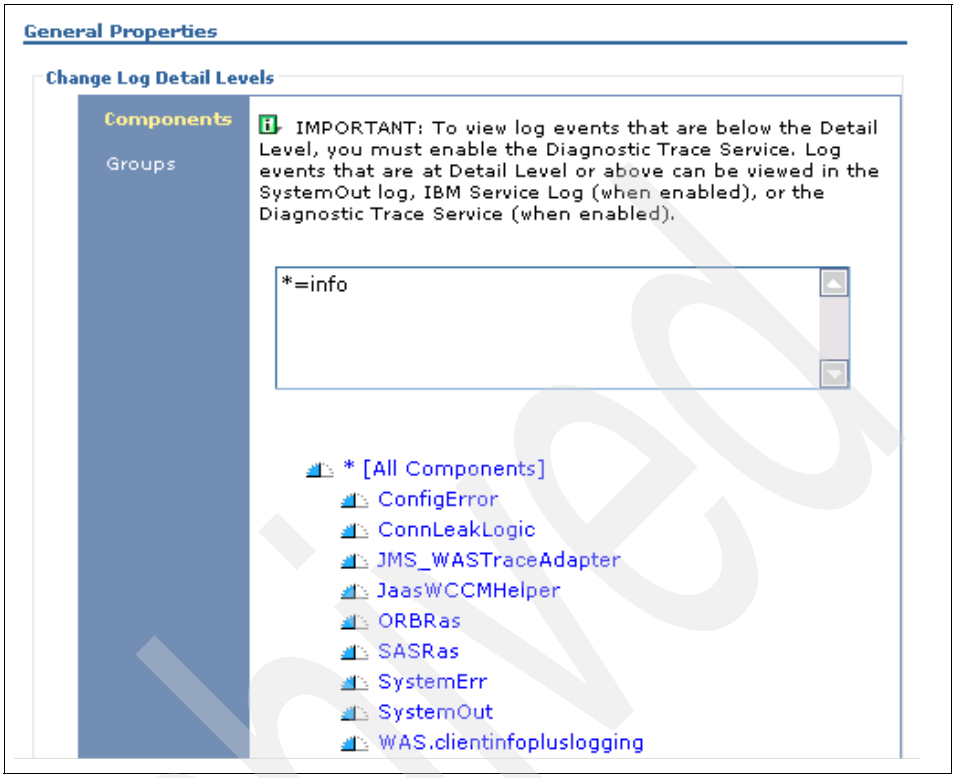

Figure 3 Changing the log detail level in the administrative console

Components can be any WebSphere Application Server packages or classes. Groups are predefined sets of packages and classes that are useful for troubleshooting a particular component.

The default log detail level is `*=info`. The log detail levels in WebSphere Application Server V6 are configured differently than the trace specifications in V5 and V5.1. If you use a V5 style trace specification as the log detail level in V6, it is mapped to the *most similar* V6 log detail level. However, to ensure that the correct tracing is enabled, we recommend only using the V6 log detail levels.

You can get a complete overview of the V6 log detail levels and how they relate to the V5 and V5.1 trace specifications in the *Log level settings* section of the WebSphere Information Center at:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/utrb_loglevel.html`

The MustGather documents for WebSphere Application Server components discuss which specific log detail level to set for different types of problems. It is a good idea to record the log detail level for different types of problems in your diagnostic data collection plan. When setting the log detail level, you should set the level to `all` in almost all cases. You should also include `*=info` in the beginning of the log detail level so that informational logging is enabled for components that are not being traced. For example, in order to enable a trace for the J2C connection manager component, you would set the log detail level to:

```
*=info:WAS.j2c=all
```

You can specify as many components and groups as you wish. You can view the components and groups that can be traced in the administrative console. However, the more components and groups you trace, the larger the trace output will be. It is a good idea to decide on a specific log detail level before enabling a trace.

After enabling the trace, clear out any old trace files from the *<WAS_install_root>*/profiles/*<profile>*/logs/*<server>* directory. Then, if you configured the trace properties and log detail level properties on the Configuration tab, restart the application server or process that you want to trace. If you configured the trace properties and log detail level on the Runtime tab, the tracing starts immediately without restarting the process. At this point, you can reproduce the problem and then disable tracing after you have reproduced it. This helps ensure that the trace file does not grow too large and makes it easier to find the time at which the problem occurs in the trace.

The trace output contains all of the messages that are also written to the SystemOut.log as well as the trace events. The trace events are in the following format:

```
[7/12/05 16:13:10:379 EDT] 00000032 DSConfigurati >  getPooledConnection Entry
```

Each entry can be deciphered as follows:

- Time stamp

  In the example, the time stamp is `[7/12/05 16:13:10:379 EDT]`.

  The time stamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (for example MM/DD/YY), 24-hour time with millisecond precision, and a time zone.

- Thread ID

  In the example, the thread ID is `00000032`.

  The thread ID is an eight-character hexadecimal value generated from the hash code of the thread that issued the trace event.

- Short name

  In the example, the short name is `DSConfigurati`.

  The short name is the abbreviated name of the component that issued the trace event. This is typically the class name of a WebSphere Application Server component, and would be some other identifier for applications.

- Event type

  In the example, the event type is a greater than symbol (>).

  The event type is a one character field that indicates the type of the trace event. The possible values are:

  - > - indicates the entry of the specified method name
  - < - indicates the exit of the specified method name
  - 1 - a trace entry of type fine or event
  - 2 - a trace entry of type finer
  - 3 - a trace entry of type finest, debug, or dump
  - Z - a placeholder to indicate that the trace type was not recognized

  The example indicates that the getPooledConnection method is entered.

- Class name

  The class name is an optional part of the trace entry. It indicates the class that generated the trace event. In the example, the class name does not appear.

- Method name

  The method name is another optional part of the trace entry. It indicates the method that generated the trace event. In the example, the method name is getPooledConnection.

- Text message

  In the example, the message is Entry.

The text message is the data that is written to the trace output file. It is meant to provide advanced debugging and troubleshooting information.

- Parameters

  Optionally, parameters can also be included in the trace entry. In the example, there are no parameters.

Collector tool

The WebSphere Application Server collector tool is a script that can be found in the *<WAS_install_root>*/bin directory (collector.bat or collector.sh). Running the script produces a Java™ archive (jar) file that contains all of the logs and XML configuration files from your WebSphere Application Server installation, as well as operating system information, Java information, and data on whether the software prerequisites were met and their levels.

You should always run the collector tool under the root or administrator user ID, and it must be run from a directory outside of the *<WAS_install_root>* directory. The resulting jar file is created in the current directory and the file name is *<host>*-*<cell>*-*<node>*-*<profile>*-WASenv.jar. For example:

```
C:\tmp\ServerHost1-Cell01-CellManager01-Dmgr01-WASenv.jar
```

The resulting jar file is very useful to the WebSphere Application Server support team and any others who are involved in the problem determination process. It allows them to view quickly your WebSphere Application Server configuration and see any errors or exceptions that have occurred.

You can find more information about the collector tool in the following WebSphere Information Center items:

- *Gathering information with the Collector tool*

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_ct.html

- *Running the collector tool*

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ttrb_runct.html

- *Analyzing collector tool output*

  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_readct.html

First Failure Data Capture (FFDC)

WebSphere Application Server V6 includes a feature called First Failure Data Capture (FFDC). The FFDC feature runs in the background and collects events and errors that occur during WebSphere Application Server runtime. The information that it collects are written to log files in the *<WAS_install_root>*/profiles/*<profile>*/logs/ffdc directory.

FFDC does not affect the performance of WebSphere Application Server and should not be disabled. The FFDC logs will not, most likely, be useful in your

problem determination efforts. However, they might be useful to the WebSphere Application Server support team if you open a PMR.

There are three FFDC configuration files in the *<WAS_install_root>*/properties directory. The only file that you should modify is the ffdcRun.properties file. You can add the ExceptionFileMaximumAge property to the file. This property specifies the number of days that an FFDC log remains in the *<WAS_install_root>*/profiles/*<profile>*/logs/ffdc directory before it is deleted. As part of your diagnostic data collection plan, you might want to modify the ExceptionFileMaximumAge property to ensure that the FFDC files remain on your system for a certain time period. You should not modify any other properties unless you are asked to do so by the WebSphere Application Server support team.

You can find more information about the FFDC feature in the WebSphere Information Center item *First failure data capture* at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_ffdc.html

Other logs

The following logs are not always useful for problem determination, but you might find that on occasion they will be required.

Process (native) logs

Native code running in a WebSphere Application Server process can write data to the process logs (also called native logs). Native code is non-Java code, typically found in files with .dll, .exe, and .so extensions. The process logs are named native_stdout.log and native_stderr.log. They are located in the *<WAS_install_root>*/profiles/*<profile>*/logs/*<server>* directory.

The only configuration that is possible for the process logs is changing the directory location or file names for the logs. You can do this in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the WebSphere Application Server process.
3. Select **Process Logs**.

You can find more information about the process logs in the WebSphere Information Center item *Process logs* at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_stdlogs.html

Service log (activity.log)

The service log is more commonly known as the activity.log and is found in the *<WAS_install_root>*/profiles/*<profile>*/logs directory. There is only one activity.log for each node. WebSphere Application Server runtime events are logged to the activity.log. It is written in binary format, so it cannot be viewed in a text editor. The main purpose of the activity.log is that it can be viewed with the Log Analyzer tool, is a graphical user interface that displays the events from the activity.log and uses a symptom database to analyze the events and diagnose problems.

You can find more information about the Log Analyzer in the WebSphere Information Center item *Log Analyzer* at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/ctrb_jfla.html

It is also possible to view the events in the activity.log outside of the Log Analyzer by using the showlog script in the *<WAS_install_root>*/bin directory.

You can find details about the usage of the **showlog** script in the WebSphere Information Center item *Showlog Script* at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_showlog.html

You can configure properties of the activity.log in the administrative console:

1. Select **Troubleshooting** → **Logs and Trace**.
2. Select the WebSphere Application Server process.
3. Select **IBM® Service Logs**.

   You can select whether to enable or disable the activity.log, choose the directory location and file name, set the maximum file size, and select what types of messages will be logged.

You can find more information about the service log in the WebSphere Information Center item *Viewing the service log* at:

http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/rtrb_viewsvclog.html

Installation logs

When you have a problem installing WebSphere Application Server V6, you might need to view the following logs to determine the failure causes.

- *<WAS_install_root>*/logs/log.txt

  This log file records the installation status

- *<WAS_install_root>*/profiles/*<profile>*/logs/pctlog.txt

  This log file records the profile creation status

- *<WAS_install_root>*/profiles/*<profile>*/logs/ivtClient.log

  This log file records the events of install verification test

IBM HTTP Server and plug-in logs and traces

When you have a problem relating to the IBM HTTP Server or the Web server plug-in, you might need to view the logs or enable a trace. This section discusses the details about these logs and traces.

IBM HTTP Server logs

The IBM HTTP Server writes two log files: an access log that contains details of all accesses to the Web server and an error log that contains details of any errors. The default location of the logs is as follows:

- Windows®
  - Access log: *<WAS_install_root>*\logs\access.log
  - Error log: *<WAS_install_root>*\logs\error.log
- UNIX®
  - Access log: *<WAS_install_root>*/logs/access_log
  - Error log: *<WAS_install_root>*/logs/error_log

Web server plug-in logs

The plug-in also writes its own log, which you can find in the Web server plug-in install directory path. The log file that you are looking for is under another directory structure, named for the logical Web server as defined in the WebSphere configuration.

You can find the location of the log file by first looking at the Web server configuration. This refers to the plug-in configuration file as shown in Example 1 on page 15. The plug-in configuration file then tells you where the log file is as

shown in Example 1. This example also shows you where you set the amount of detail that is logged.

Example 1 Location of plug-in log file

```
<Log LogLevel="Error"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

The default setting for `LogLevel` is `Error`, but you can set it to `Trace` to collect significantly more information. Should you need to raise this problem with IBM Support, they will request a plug-in trace.

Web server plug-in trace

To get an effective trace, you need to enable as much logging as possible on the Web server. For example, you can set the logging level to capture verbose output in the IBM HTTP Server by modifying the `LogLevel` directive in the configuration file as shown:

```
LogLevel debug
```

You need to restart the IBM HTTP Server for this to take effect.

Enable trace logging in the Web server plug-in by setting the `LogLevel` directive in the plugin-cfg.xml file as shown:

```
<Log LogLevel="Trace"
Name="c:\ibm\was6\plugins\logs\webserver1\http_plugin.log" />
```

You do not need to restart the IBM HTTP Server for this change to take effect.

Tip: The plug-in trace generates significant amounts of data. Make your test as specific as possible, and run it in isolation to reduce the number of lines generated.

Network trace

In rare cases, you might need to use a network protocol analyzer that allows you to capture an iptrace. This tool can help you to determine where the problem lies. WebSphere Application Server does not supply such a tool. However, there are third-party tools available (for example, Ethereal from `http://www.ethereal.com/`).

System management logs

When you have a system management problem, you might need to view certain logs or enable a trace. This section discusses the details about these logs and traces.

Output from wsadmin

Messages from wsadmin are written to the wsadmin.traceout log file:

<WAS_install_root>/profiles/*<profile>*/logs/wsadmin.traceout

You can also increase the amount of data that is logged to this file by tracing the wsadmin utility. To do so, update the following file:

<WAS_install_root>/properties/wsadmin.properties

Uncomment the following line:

```
com.ibm.ws.scripting.traceString=com.ibm.*=all=enabled
```

Note that the information that is logged is of limited use because wsadmin calls MBeans in the application server that is running the administrative console application. So, you usually need to trace that application server as well.

Management scripts

You can manage WebSphere Application Server services using the supplied management scripts. For example, each WebSphere Application Server installation has a script to start an application server, a script to stop an application server, and a script to show you the status of all application servers defined in a profile. Each of these scripts writes its own log file into the server's logs directory. For example, the stopServer script writes stopServer.log into the logs directory:

<WAS_install_root>/profiles/*<profile>*/logs/*<server>*/stopServer.log

Profile management logs

The profile creation and management tool wasprofile writes messages to the profile independent logs directory, that is:

<WAS_install_root>/logs/wasprofile/*<profile>*.log

This log file is in XML format.

The Java graphical interface that is used to create a profile simply calls the **wasprofile** command after collecting the information needed. By default, it does not write a log, but you can pass it a log parameter as shown:

```
pctWindows -is:log c:\temp\pct.log
```

WebSphere Rapid Deployment logs

The WebSphere Rapid Deployment tool works on a directory that you create and pass to WebSphere Rapid Deployment in the WORKSPACE environment variable. It logs Eclipse messages into two separate files within this directory:

- *<workspace>*/.metadata/.log
- *<workspace>*/*project*/.metadata/.log

In a manner similar to other WebSphere Application Server utilities, WebSphere Rapid Deployment calls MBeans on the application server. The application server logs can help you resolve a problem with WebSphere Rapid Deployment. There is no way to trace the WebSphere Rapid Deployment utility. However, you can trace the application server as described in “Tracing” on page 6.

Summary of logs

Table 1 on page 18 shows a summary of the WebSphere logs.

Note: In this table:

- *<WAS_install_root>* represents the installation root for WebSphere Application Server, for example:

  ```
  c:\WebSphere\Appserver
  ```

- *<profile_home>* represents the root directory for a specific WebSphere Application Server profile, for example:

  ```
  <WAS_install_root>/profiles/<profile>
  ```

- *<ihs_install>* represents the installation directory for the IBM HTTP Server, for example:

  ```
  c:\IBM HTTP Server
  ```

Table 1 Log file summary

Tasks	Logs	Format / tools
Installation tasks		
WebSphere Installation	*<WAS_install_root>*/logs/log.txt	text
IBM HTTP Server installation	*<ihs_install>*/ ► ihsv6_install.log ► gskitInstall.log	text
Sample and IVT application installation: ► DefaultApplication ► ivtApp ► query ► PlantsByWebSphere ► SamplesGallery	*<profile_home>*/logs/ ► defaultapp_config.log ► defaultapp_deploy.log ► ivt_config.log ► query_config.log ► samples_config.log ► samples_install.log	text
WebSphere system application installation: ► filetransfer ► filetransferSecured ► ManagementEJB ► SchedulerCalenders ► adminconsole	*<profile_home>*/logs/ ► filetransfer_config.log ► mejb.log ► scheduler.cal_config.log ► webui_config.log	text
Profile tasks		
Profile creation wizard	*<profile_home>*/logs/pctLog.txt	text
	<WAS_install_root>/logs/waspro file/wasprofile_create_*<profile>*.l og.	XML
Application and system tasks		
Application print() and println() JVM System.out and System.err streams	*<profile_home>*[2]/logs/*<server>*/: ► SystemOut.log[1] ► SystemErr.log[1]	Text. administrative console: **Troubleshooting → Logs and Trace → *<server>* →JVM Logs**

[1] Configurable
[2] *<profile_home>* can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, *<server>* is “nodeagent”. If the profile is for a deployment manager, *<server>* is “dmgr”

Tasks	Logs	Format / tools
WebSphere processes	Process logs at *<profile_home>*[2]/logs/*<server>*/: ► native_stderr.log[1] ► native_stdout.log[1]	Text (the server must be stopped to view with a text editor) administrative console: **Troubleshooting → Logs and Trace → *<server>* →Process Logs**
WebSphere System.out stream + messages that contain extended service information	IBM service log (aka activity log) at *<profile_home>*/logs/activity.log[1]	Binary format. View with Log Analyzer. Showlog tool can convert the contents to a text format.
Operational tasks		
Start / stop an application server.	*<profile_home>*/logs/*<server>*/: ► SystemOut.log[1] ► SystemErr.log[1] If using **startServer *<server>* -trace**, see *<profile_home>*/logs/*<server>*/: ► startServer.log ► stopServer.log	text
Start / stop a deployment manager	*<profile_home>*/logs/dmgr/: ► SystemOut.log[1] ► SystemErr.log[1] *<profile_home>*/logs/dmgr/: ► startServer.log ► stopServer.log	text
Start / stop a node agent	*<profile_home>*/logs/nodeagent/: ► SystemOut.log[1] ► SystemErr.log[1] If using **startNode -trace**, see *<profile_home>*/logs/nodeagent/: ► startServer.log ► stopServer.log	text

[1] Configurable
[2] *<profile_home>* can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, *<server>* is “nodeagent”. If the profile is for a deployment manager, *<server>* is “dmgr”

Tasks	Logs	Format / tools
Start / stop a cluster	When you start or stop a cluster, that action is taken on each server. The logging is the same as though you started or stopped each server.	
Configuration tasks		
Adding a node to a cell	*<node_profile_home>*/logs/: ► addNode.log ► runAddNode.log	text

[1] Configurable
[2] *<profile_home>* can represent the location of the profile for an application server, node agent, or deployment manager. If the profile is for a node agent, *<server>* is “nodeagent”. If the profile is for a deployment manager, *<server>* is “dmgr”

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

This document created or updated on September 15, 2005.

Send us your comments in one of the following ways:
- Use the online **Contact us** review redbook form found at:
  ibm.com/redbooks
- Send your comments in an email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HZ8 Building 662, P.O. Box 12195
  Research Triangle Park, NC 27709-2195 U.S.A.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

@server®	Redbooks (logo) ™	Redbooks™
@server®	IBM®	WebSphere®

The following terms are trademarks of other companies:

Java, JVM, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, or service names may be trademarks or service marks of others.