IBM Software Group

SW5706
Problem determination techniques

business on demand.

Updated September 20, 2007

4.0

This presentation will focus on problem determination techniques with WebSphere Application Server V6.

Unit objectives

After completing this unit, you should be able to:

- Gather information for problem determination:
  - Search for information in the WebSphere Knowledge Base and the WebSphere Support page
  - Use product Information Centers
  - Use MustGather documents
  - Use the Troubleshooting Guide
- Build a detailed low-level timeline of events for deep analysis
- Identify and describe the main problem determination artifacts: logs, traces, dumps, PMI data, and so on
- Enable basic tracing of the server and HTTP plug-in
- Check product versions and patch levels
- Locate and interpret WebSphere FFDC logs

After completing this unit, you should be able to gather pertinent problem determination data.

WebSphere Knowledge Bases search

- The WebSphere Knowledge Bases provide a good starting point for gathering information
- Choose keywords to research the problem in available knowledge sources:
  - Error codes, exception names, and so on from the most promising symptoms
  - If no explicit error, use high-level problem description
    - See list of problem categories from the IBM Support Web site
- Research in IBM Support Assistant or on the WebSphere Application Server Support Web page
  - Will automatically search product Information Center for error codes, and so on
  - Support page address: http://www.ibm.com/software/webservers/appserv/was/support/

The first step in problem determination is knowing where to look to find answers. The WebSphere Knowledge Bases provide a good starting point for gathering information. Search using keywords based on any explicit error codes, or use a high-level problem description to begin your search. The IBM Support Assistant and the WebSphere Application Server Support Web Page are excellent resources. The IBM Support Assistant will be covered in another presentation.

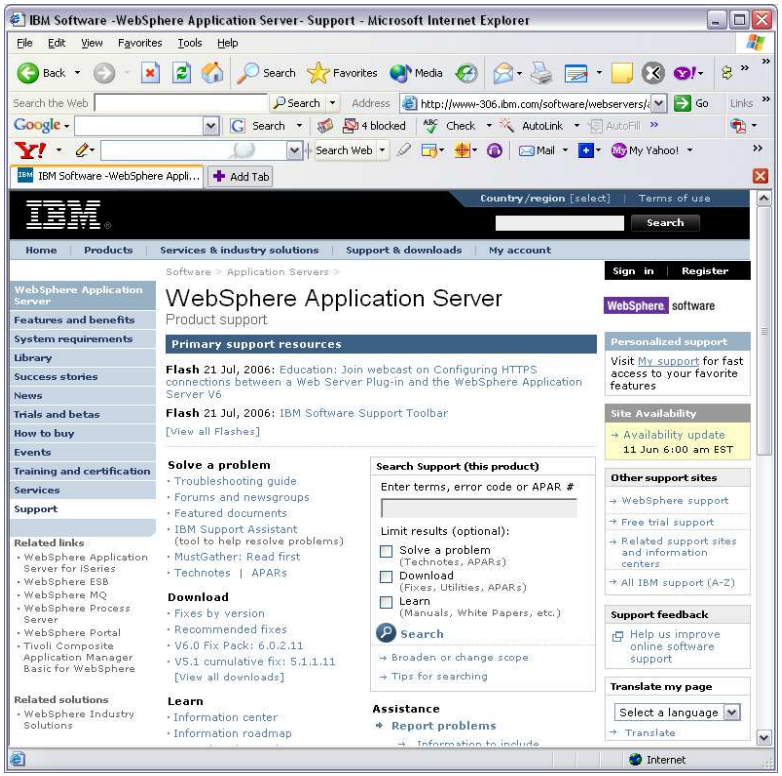
WebSphere Knowledge Bases and Support page

- All the problem determination data is organized into a set of predefined problem categories or “components”
  - Currently 44 categories for WebSphere Application Server: “100% CPU Usage”, “Admin Console”, “Classloader”, “Crash”, and so on
  - Also used to manage most support processes and other problem determination assets
- Library of Technotes and other articles
  - Maintained by a specialized knowledge engineering team
  - Includes known problems, APARs, common questions, troubleshooting instructions for many specific problems, problem determination tools, and so on
- One *MustGather* document for each problem category (few exceptions)
  - Provides instructions on how to start troubleshooting that problem, and what information to provide to support if opening a PMR
- Extensive search facility

The WebSphere Support page provides links to the MustGather recommendations, which are organized into predefined problem categories such as “crash” and “out of memory”. A library of Technotes and other timely featured documents is also accessible from this page.

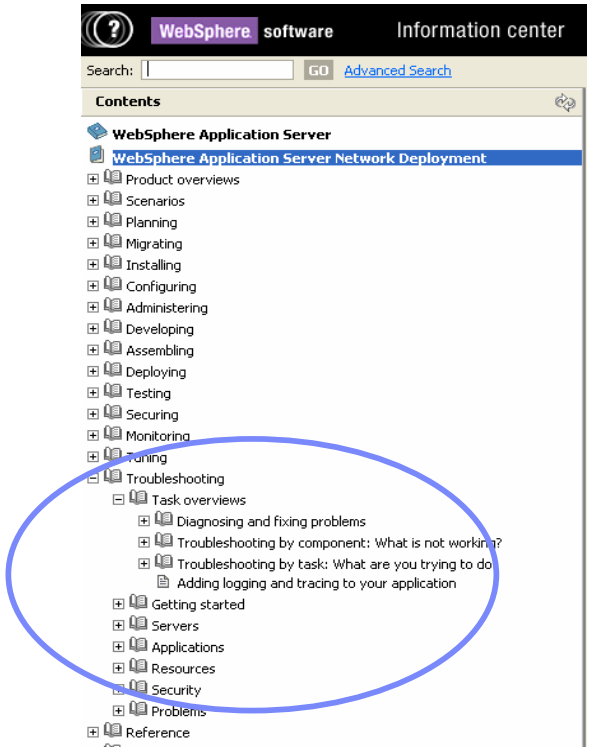
Searching on the Support Page

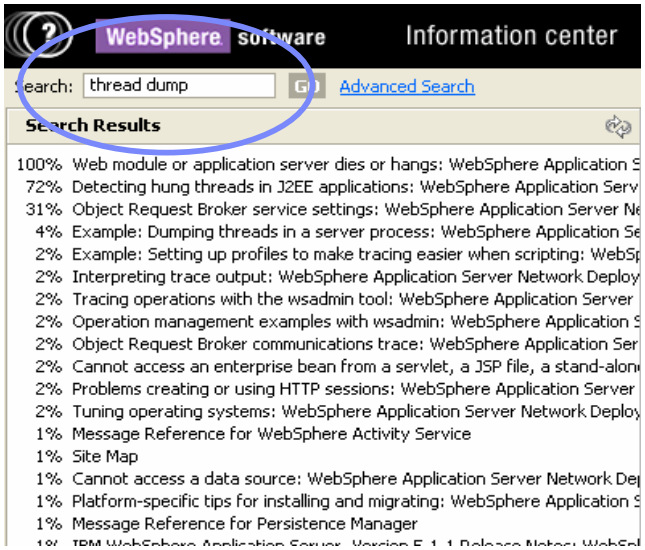
- Try using ISA (covered in another unit)
- Search on the exception or error message you are seeing
- Search on APAR or Fixpack
- Search for problem determination tools
- Search for *MustGather* documents



Searching for information on the WebSphere Application Server Support Page provides many avenues. A search can be made on a specific exception or error. Searching on APARs, Fixpacks, problem determination tools, and MustGather documents can also be commenced on this page.

Searching the Information Center





The WebSphere Information Center can be searched using keywords, or the hierarchical selection of topics may be walked. A troubleshooting section provides examples of many problem solving scenarios.

Product components and MustGather

Main MustGather document page: http://www.ibm.com/support/docview.wss?rs=180&uid=swg21145599

- Use the collector tools to capture general information for each host machine

Components	Collector Tool		
	Upgrade Policies		
	IBM Education Assistant		
100% CPU Usage		read	
Administrative Console (all non-scripting)	learn		read
Administrative Scripting Tools (for example: wsadmin or ANT)	learn		read
Application Client		read	
Application Server Toolkit (AST)			
Classloader	learn		read

A description of components:

http://www.ibm.com/support/docview.wss?rs=180&context=SSEQTP&uid=swg21165548

Component	Description
100% CPU Usage	Includes multiple product categories. 100% CPU usage due to a WebSphere Application Server process.
Administrative Console (all non-scripting)	Any problems associated with using the Administrative Console, not including scripting (wsadmin). Does not include problems that occur after changing a setting that causes a change in behavior.
Administrative Scripting Tools (for example, wsadmin or ANT)	Problems experienced performing Administration tasks using command line tools

The MustGather documentation is divided into predefined problem categories, such as “100% CPU Usage”, “Classloader”, and “Crash”. A link to a description of these problem categories is also provided. Using the collector tool is an easy way to gather the majority of the requested documentation.

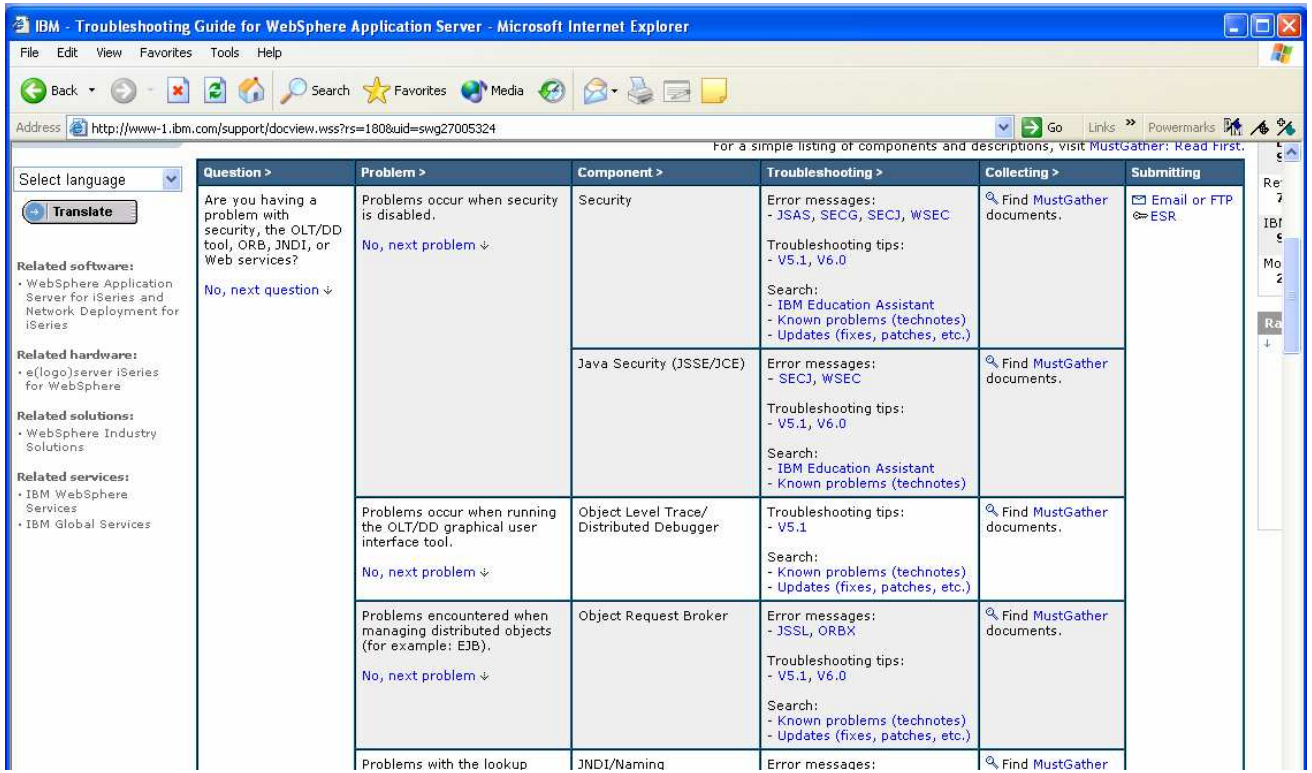
MustGather example: crash on Linux

1 - 8 of 8 items found		Modified date
[1]	MustGather: Crash on Linux **MustGather** for problems with a WebSphere® Application Server crash on Linux. Gathering this information before calling IBM support will help familiarize you with the troubleshooting process and save you time. [More items like this found in Distributed Application & Web Servers]	2004-02-25

- Use operating system-specific MustGather documents, when appropriate
- Product components include symptoms like “Crash” that span product components
- Updated frequently to help resolve problems more quickly

Here is an example of a link to a MustGather document for a specific type of problem on a specific platform.

Troubleshooting Guide



The Troubleshooting Guide helps you get started on the troubleshooting process. It takes you through the process of identifying which component is causing the problem, finding the appropriate troubleshooting information, then collecting any necessary MustGather information, and finally submitting a problem to IBM Support.

WebSphere Application Server logs

- JVM logs: created by redirecting the System.out and System.err streams of the JVM to independent log files
  - One set of JVM logs for each application server and all of its applications located by default in the following directory: *install_root*/profiles/*profile_name*/logs/*server_name*
- Process logs: contain two output streams (stdout and stderror) that are accessible to native code running in the process
  - One set for each application server
- IBM service log: contains both the WebSphere Application Server messages that are written to the System.out stream and some special messages that contain extended service information that is normally not of interest, but can be important when analyzing problems
- The HTTP server plug-in maintains a special log

The various logs of the WebSphere Application Server are sources of valuable information for problem determination. The JVM logs are created by redirecting the System.out and System.err streams of the JVM to independent log files. Process logs contain two output streams, stdout and stderr, which are accessible to native code running in the process. The IBM service log contains WebSphere Application Server messages and extended service information in a binary format which requires a special tool, such as the Log Analyzer, to view. The HTTP server plug-in also maintains a log.

Log Files and Locations

- The destination and names for the log files are configurable. The default location is:

```
<was_root>\profiles\<profile_name>\logs\<server_name>
```

- Log Files:
  - **SystemOut.log** and **SystemErr.log** - Standard JVM output and error log
  - **startServer.log** and **stopServer.log**
    - Startup and shutdown of the Application Servers
  - **activity.log** - Events that show a history of activities
    - Use Log Analyzer to read output from this file
  - **trace.log** – Output from diagnostic trace
    - Destination and name are configurable
  - **http_plugin.log** – Not in <was_root>
    - Location: `<plugin_root>\logs\<webserver_name>`
  - **native_stdout.log** and **native_stderr.log**

The location and names for most log files are configurable. SystemOut.log and SystemErr.log are the default names for the JVM logs. They contain server, as well as, user program information.

startServer.log and stopServer.log contain information logged by the server as it starts-up and shuts-down. The activity log shows a history of activities. The trace log holds output from activated diagnostic traces.

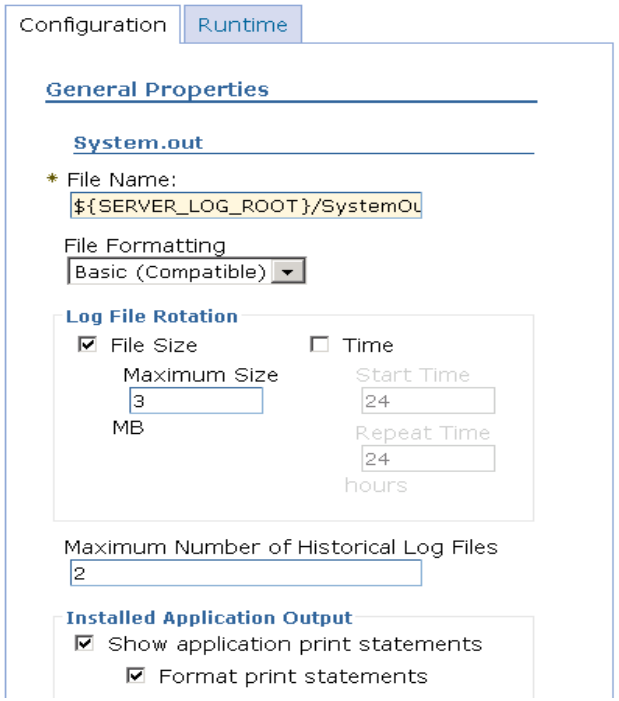
Viewing logs

- JVM logs:
  - Click **Troubleshooting > Logs and Trace** in the administrative console navigation tree, or
  - Navigate to the *profile_root*/logs/*server_name* directory on the machine where logs are stored, and open SystemOut.log or SystemErr.log in a text editor
- Process logs:
  - *Open profile_root*/logs/*server_name*/native_stderr.log or *profile_root*/logs/native_stdout.log in a text editor
- IBM service log:
  - Use AST to view, or
  - Use the Log Analyzer tool

Logs can be viewed in a variety of manners. The JVM logs may be viewed within the administrative console, or they may be opened from the file system by a text editor. The process logs may be viewed by a text editor. The service log and activity logs are in a binary format, and must be viewed with the Application Server Toolkit or the Log Analyzer tool.

Configuring JVM Logs

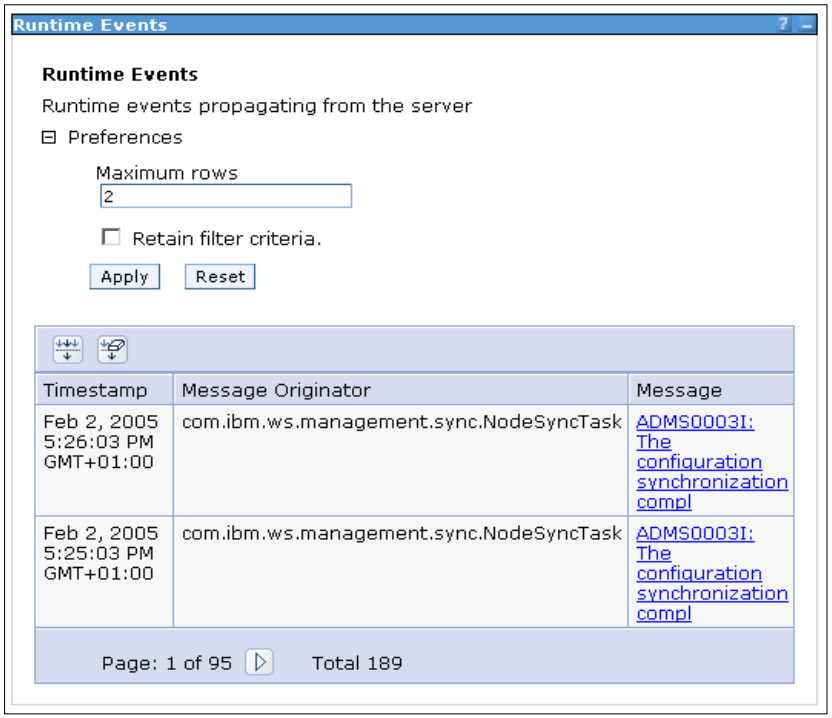
- Troubleshooting → Logs and Trace > <server_name> → JVM Logs
- *Alternative*: Servers -> Application servers → <server_name> → Logging and Tracing → JVM Logs
- System.out and System.err logs can be configured from this page
- Logs are self-managing
  - Can roll over based on time or file size
  - Number of historical log files is configurable
- To view logs through the console, use the Runtime tab



The JVM logs can be managed from the administrative console. File names, locations, and roll-over behavior may be specified.

Viewing messages in the console (1 of 2)

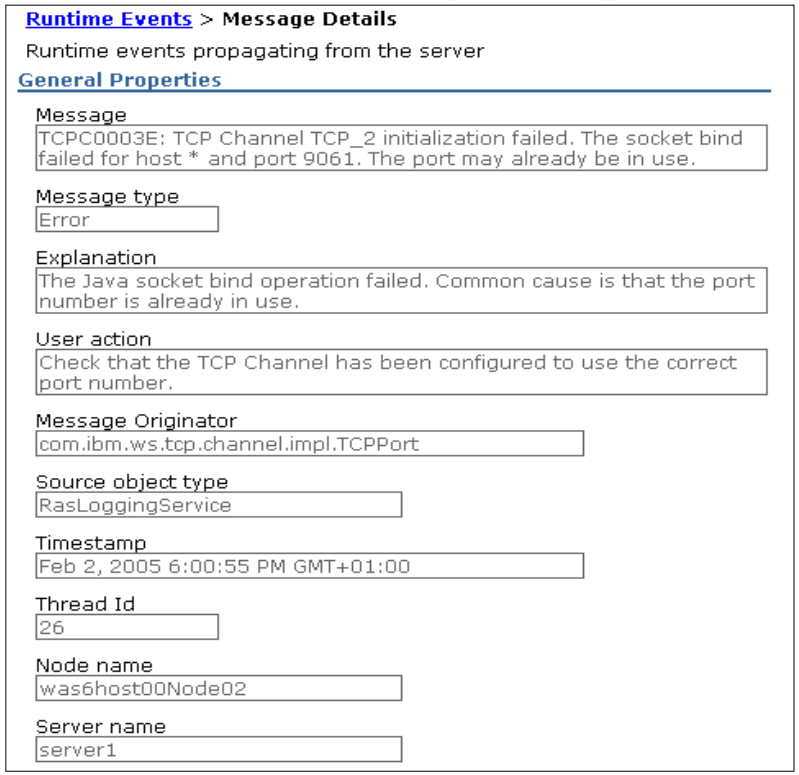
- Runtime events are grouped by severity: Error, Warning, Information
- View: Troubleshooting -> Runtime Messages -> Click:
  - Error
  - Warning
  - Information



While viewing Runtime messages, first select the Error, Warning or Information category links (a count of zero means nothing is available). Then the details for the selected category are shown. Selecting one of these links will provide detail information. Note that you may have multiple pages of messages, the button on the bottom of the page will allow you to see them all.

Viewing messages in the console (2 of 2)

- Runtime events details



Message detail information is displayed on the detail screen for the event for you to resolve the problem with user action.

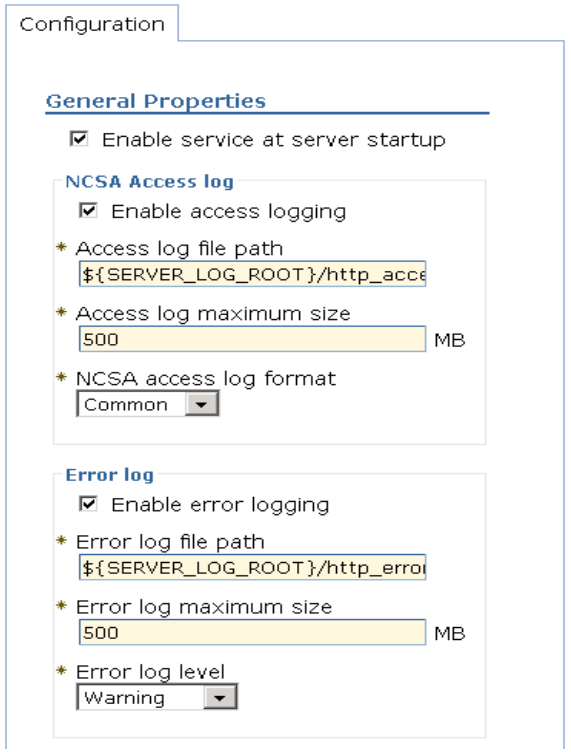
HTTP plug-in logs and tracing

- Click **Servers > Web Servers >** *web_server_name* **Plug-in Properties** > Configuration tab > Plug-in logging to edit fields
- Default location:
  plugins_root/logs/*web_server_name*/http_plugin.log
- Set the Log level to Trace to trace all the steps in the request process (caution: this produces a lot of output)

The configuration for the logs and tracing of the HTTP plug-in is managed in the administrative console. A variety of trace settings are available.

Embedded HTTP Server Logs

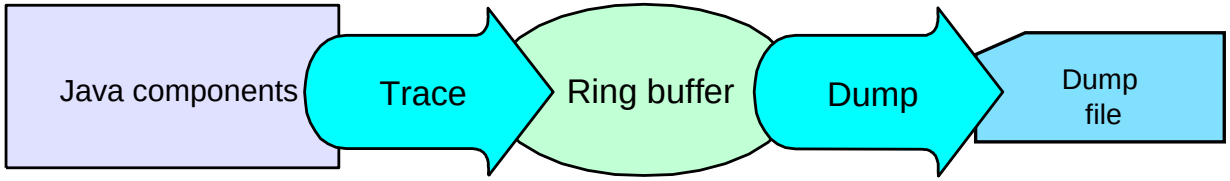
- Administrative Console panels for configuring embedded HTTP server logs (access and error)
- From main application server panel, click HTTP Error and NCSA Access Logging
- Access and error logs can be controlled separately
- When maximum file size is reached, oldest entries are pruned



Logging can also be activated for the embedded HTTP server. The access and error logs can be enabled and controlled separately.

Traces (1 of 2)

- Trace files show the time and sequence of methods called by WebSphere Application Server base classes, and you can use these files to pinpoint the failure
- Trace can be started
  - While server is running using Runtime Diagnostic Trace
  - When server is started using Configuration Diagnostic Trace
- Trace output can be directed to:
  - Memory ring buffer - dumped after trace stops
  - File
- Trace has a significant impact on performance
  - Enable temporarily for problem determination
  - Trace to file is slower than trace to memory ring buffer Runtime tab



Trace output allows administrators to examine processes in the application server and diagnose various issues. Trace output can be directed either to a file or to an in-memory circular buffer. If trace output is directed to the in-memory circular buffer, it must be dumped to a file before it can be viewed. In all cases, trace output is generated as plain text in either basic, advanced or log analyzer format as specified by the administrator. The basic and advanced formats for trace output are similar to the basic and advanced formats that are available for the JVM message logs.

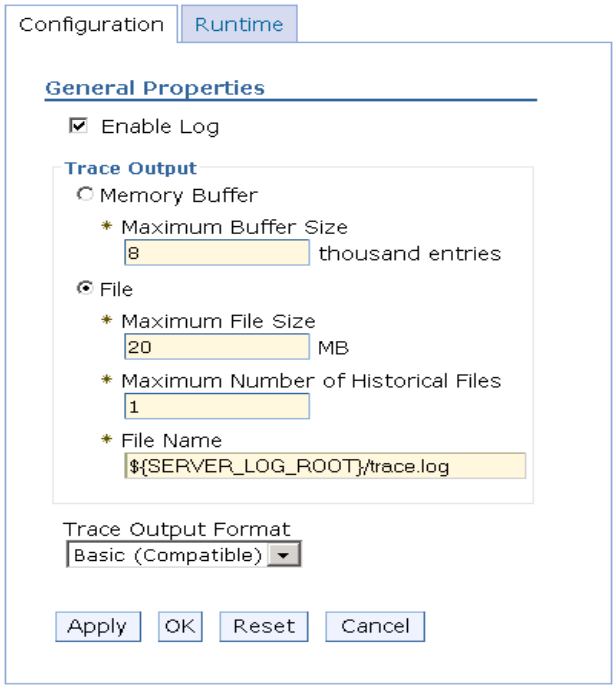
Traces (2 of 2)

- The procedure for using trace is as follows:
  1. Configure an output destination to which trace data is sent
  2. Enable trace for the appropriate WebSphere Application Server or application components
  3. Run the application or operation to generate the trace data
  4. Analyze the trace data or forward it to the appropriate organization for analysis
- Click **Troubleshooting > Logs and Trace >** *server_name* in the administrative console

The procedure for using trace is as follows: configure an output destination for the trace data, enable trace for the appropriate WebSphere Application Server or application components, run the application or operation to generate the trace data, then analyze the trace data or forward it to the appropriate organization for analysis.

Enabling Trace

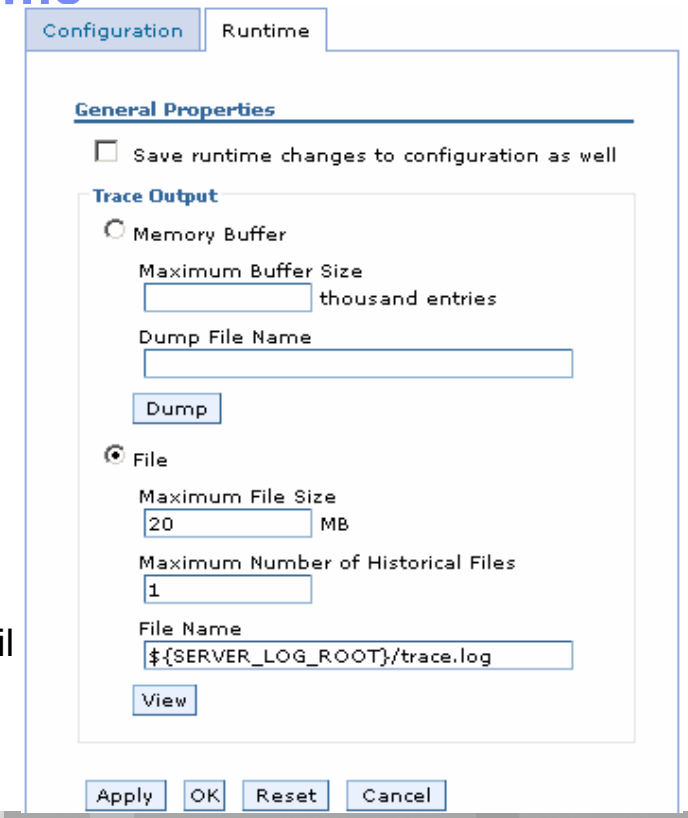
- Enabled by default (no trace output)
- Troubleshooting → Logs and Trace → <server_name> → Diagnostic Trace
- Enable Log checkbox enables tracing
- Configurable output
  - Memory buffer or file
- The location for entering the trace string is on the Log Detail Level page
- **Note**: Change Log Detail Level to get trace output



The “Enable Log” checkbox on the configuration tab is enabled per default. Because of the Log Detail Level is set to ***=info** there is not trace output. The trace level string is specified on the Log Detail Level page.

Trace Dump and Runtime

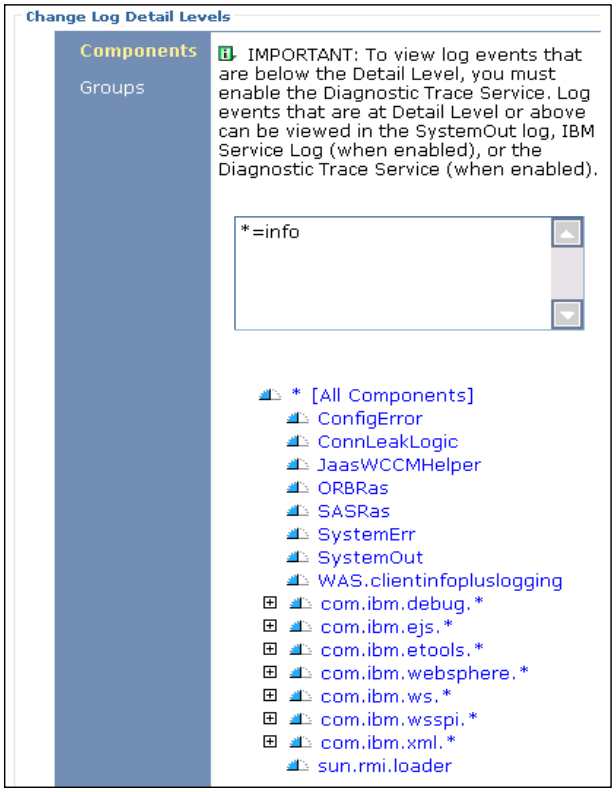
- View and dump available in the Runtime tab of diagnostic trace
- Application Server Toolkit (AST) can be used to analyze trace output, but you may prefer to use your favorite editor; advanced users may want to use a tool like ProTrace
- Before you can view or dump trace you need to specify log detail level



A temporary trace configuration for the dump and view facilities can be set on the runtime tab. But before you can view or dump the trace, you need to specify the log detail level.

Setting the Log Detail Level

- **Logs and Tracing → Change Log Detail Level**
- Log detail level affects tracing *and* regular logging
  - Setting levels below **info** reduces the amount of data in logs
  - ***=off** disables logging altogether
- Trace levels (**fine**, **finer**, **finest**) do not appear in the trace file unless logging is enabled
- Log string can be typed in or set using the graphical menu
- Default is ***=info**
- User-created applications can be instrumented too, and be included in the trace output



Log Levels control which events are processed by Java logging. WebSphere Application Server controls the levels of all loggers in the system. The level value is set from configuration data when the logger is created and can be changed at run time from the administrative console. Trace information, which are events at levels Fine, Finer and Finest, can only be written to the trace log. Therefore, if you do not enable diagnostic trace, setting the log detail level to Fine, Finer or Finest will not have an effect on the data that is logged.

Trace formats

- **Basic format--**Trace events displayed in basic format use the following format:
  - `<timestamp><threadId><shortName><eventType>[className][methodName] <textmessage> [parameter 1] [parameter 2]`
  - **`[4/20/06 16:04:44:429 EDT] 0000000a UserRegistryI 3   user uid=wpsbind,cn=users,dc=ibm,dc=com password checks ok`**
- **Advanced formats--**Trace events displayed in advanced format use the following format:
  - `<timestamp><threadId><eventType><UOW><source=longName>[className] [methodName] <Organization><Product><Component>[thread=threadName] <textMessage>[parameter 1=parameterValue][parameter 2=parameterValue]`
  - **`[4/20/06 22:31:09:316 EDT] 0000000a  1 UOW=null source=com.ibm.ws.security.core.distSecurityComponentImpl org=IBM prod=WebSphere component=Application Server thread=[Thread-1] SECJ0240I: Security service initialization completed successfully`**

You can specify one of three levels for trace output:

Basic, or compatible, preserves only basic trace information. Select this option to minimize the amount of space taken up by the trace output.

Advanced format preserves more specific trace information. Select this option to see detailed trace information for use in troubleshooting and problem determination.

Log analyzer trace format allows you to open trace output using the Log Analyzer. Log Analyzer format is useful if you are trying to correlate traces from two different server processes, because it allows you to use the merge capability of the Log Analyzer.
The descriptions of the trace fields are on the next two foils.

Trace fields (1 of 2)

- **TimeStamp**
  - The timestamp is formatted using the locale of the process where it is formatted. It includes a fully qualified date (YYMMDD), 24 hour time with millisecond precision and the time zone.
- **ThreadId**
  - An 8 character hexadecimal value generated from the hash code of the thread that issued the trace event.
- **ThreadName**
  - The name of the Java thread that issued the message or trace event.
- **ShortName**
  - The abbreviated name of the logging component that issued the trace event. This is typically the class name for WebSphere Application Server internal components, but may be some other identifier for user applications.
- **LongName**
  - The full name of the logging component that issued the trace event. This is typically the fully qualified class name for WebSphere Application Server internal components, but may be some other identifier for user applications.
- **EventType**
  - A one character field that indicates the type of the trace event. Trace types are in lower case.

The following is a description of the trace fields.

Trace fields (2 of 2)

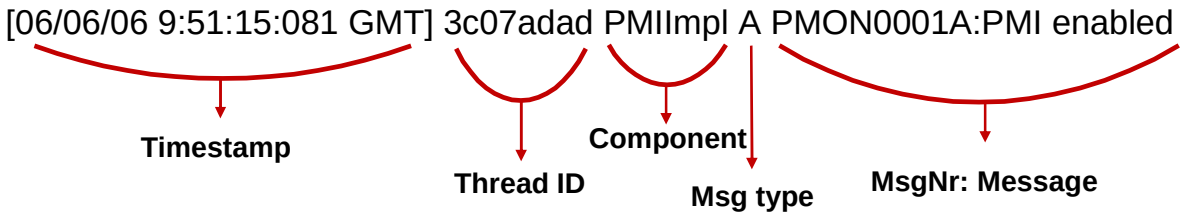
- **ClassName**
  - The class that issued the message or trace event
- **MethodName**
  - The method that issued the message or trace event
- **Organization**
  - The organization that owns the application that issued the message or trace event
- **Product**
  - The product that issued the message or trace event
- **Component**
  - The component within the product that issued the message or trace event

Here is a continuation of the description of the trace fields.

Reading a Trace File

- Timestamps give good clues:
  - Time stamps are real machine time values
  - Good when comparing traces from different processes
- Look for exceptions (search for exception from top)
  - Events preceding the exception are probable causes
  - Events after exception are recovery attempts
- Often useful to follow a single thread

>	Entry to a method (debug)
<	Exit a method (debug)
A	Audit
W	Warning
X	Error
E	Event (debug)
D	Detail (debug)
T	Terminate (exits process)
F	Fatal (exits process)
I	Information
O	Program output
C	Configuration



Search for exceptions from the top of the trace file. It is often useful to follow a single thread. A tool such as the Trace Analyzer makes this easy.

Example: low-level timeline

```
                HTTPD           Mars            Mars            Mars
                Count           Thread 1        Thread 2        Thread 3

12:00:00:***                    RESTART ---------------------------------

09:30:00:***    10

09:55:00:***    10

09:59:10:564                    > getConnection
09:59:10:943                    < getConnection

09:59:15:134                                    > getConnection
09:59:15:201                                                    > getConnection
09:59:15:456                    > getConnection

10:00:00:***    125

10:02:15:859                                    CONM6026W

10:05:00:***    192

10:10:00:***    193

10:10:01:034                                    > getConnection
10:10:01:750                    GC allocation failure (2Meg) -----------
10:10:01:900                    CRASH -----------------------------------
```


Creating a low-level timeline is a useful technique for doing “phase 1” investigation, but does not apply in every problem case. If you are looking at a log or trace file for longer than five minutes, you may want to use a chart like this to organize your analysis. It may help you to keep track of certain events and when they occurred.

Checking version levels and applying APARs

- The **versionInfo** command generates a report from data extracted from XML files in the properties/version folder. The report includes a list of changed components and installed or uninstalled maintenance packages.
- APAR: Authorized Program Analysis Report; tracks software defects reported by customers
- Download recommended fixes, and fixes by version from the WebSphere Support page
- **Fixpack installer**: is now the standard tool for installing fixes; you can find more information on this in the Information Center

The versionInfo command generates a report regarding an installed or uninstalled maintenance packages. Use this information to plan for recommended or routine maintenance. Download recommended fixes from the WebSphere support page. The fixpack installer is the standard tool for installing fixes.

FFDC

- The first failure data capture (FFDC) log file saves information that is generated from a processing failure (for example, a Java exception)
  - Captured data is saved in log files for use in analysis
    - An index file that references all of the exceptions logged by FFDC
    - An exception file for each exception type from each probe
  - Capturing FFDC data does not affect performance
  - FFDC data is collected in the <profile_root>\logs\ffdc directory
- You can configure the number of days this information is saved (afterwards, it is deleted)
- Retrieve these log files using an FTP client from any other environment
- Because the index and exception logs are text files, they can be viewed in any ASCII-capable text editor or viewer

The first failure data capture (FFDC) feature preserves the information that is generated from a processing failure and returns control to the affected engines. The captured data is saved in a log file for analyzing the problem. FFDC is intended primarily for use by IBM Service.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject= Feedback about SW5706G04_Techniques.ppt

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

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

IBM WebSphere

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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. 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 in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.