



Problem determination



Unit objectives

After completing this unit, you should be able to:

- Describe a basic approach for problem determination
- List resources for completing a problem investigation
- Locate relevant log files
- Examine log activity
- Enable tracing on specific components
- Enable high performance extensible logging (HPEL) and use the Log Viewer
- Locate other troubleshooting tools
- Use the IBM Support Assistant

Topics

- Server logs and diagnostic tracing
- Gathering diagnostic data
- Problem determination tools
- IBM Support Assistant



Server logs and diagnostic tracing





Examining server log files

- Log files are an initial source of diagnostic data
- WebSphere provides several useful logs, including:
 - JVM logs
 - Process logs
 - HTTP plug-in logs
 - Console runtime messages

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 that are located by default in the following directory:
`<profile_root>/<profile_name>/logs/<server_name>`
 - `SystemOut.log` and `SystemErr.log`
- **Process logs:** contain two output streams (`stdout` and `stderr`) which are accessible to native code that runs in the process
 - One set for each application server
 - `native_stderr.log` and `native_stdout.log`
- **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
 - One per profile (node)
 - `activity.log`



Server log files: Types and locations

- Application server log file destinations and names are configurable
 - The default location is:
`<was_root>\profiles\<profile_name>\logs\<server_name>`
- Application server log files are all text files
 - JVM logs: `SystemOut.log` and `SystemErr.log`
 - Start and stop logs: `startServer.log` and `stopServer.log`
 - Diagnostic trace log: `trace.log`
 - Process logs: `native_stdout.log` and `native_stderr.log`
- Web server plug-in log file:
 - `http_plugin.log`
 - Location: `<plugin_root>\logs\<webserver_name>`
- IBM service log is in binary format
 - In `<profile_root>\<profile_name>\logs\activity.log` directory
 - Use Log Analyzer in IBM Support Assistant to view

Configuring JVM logs

- From the administrative console, select **Troubleshooting > Logs and Trace > *server_name* > JVM Logs**
- SystemOut and SystemErr 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

Configuration **Runtime**

General Properties

System.out

* File Name:

File Formatting

Log File Rotation

☒ File Size ☐ Time

Maximum Size

MB

Start Time

Repeat Time

hours

Maximum Number of Historical Log Files

Installed Application Output

☒ Show application print statements

☒ Format print statements

Viewing runtime messages in the console (1 of 2)

- Runtime events are grouped according to severity: error, warning, information
- To view, select:
Troubleshooting > Runtime Messages >
 - Runtime Error
 - Runtime Warning
 - Runtime Information
- Runtime events are disabled by default
 - None
- Select **Info** to enable all runtime events

The screenshot shows the 'Runtime Events' console window. It has a title bar with a question mark icon. The main content area is titled 'Runtime Events' and contains the text: 'Use this page to view runtime events that propagate from the server.' Below this is a green information icon followed by text explaining that runtime events are disabled by default and that users can select a level from a list: 'Error' for only error events, 'Warning' for both error and warning events, and 'Info' for all events. A dropdown menu is open, showing the options: 'None', 'Error', 'Warning', and 'Info' (which is highlighted with a blue selection bar). Below the dropdown is a '+ Preferences' link. At the bottom of the window is a table with three columns: 'Timestamp', 'Message Originator', and 'Message'. The table currently shows 'None' in the 'Message' column and a 'Total 0' row at the bottom.

Timestamp	Message Originator	Message
		None
Total 0		

Viewing runtime messages in the console (2 of 2)

- Runtime events details include:
 - Message code and text
 - Brief explanation of the event
 - Action for the user to take
 - What server component issued the message
- Other details include:
 - Timestamp
 - Thread ID
 - Node name
 - Server name

Runtime Events ? -

[Runtime Events](#) > [Message Details](#)

Use this page to view runtime events that propagate from the server.

General Properties

Message
SRVE0255E: A WebGroup/Virtual Host to handle /PlantsByWebSphere has not been defined.

Message type
Runtime error

Explanation
Could not find a web group (web module) or virtual host to handle the request. This is an application error.

User action
Be sure the web group and virtual host is defined and deployed.

Message Originator
com.ibm.ws.webcontainer

Source object type
RasLoggingService



HTTP plug-in logs and tracing

- To configure plug-in logs and tracing from the administrative console, click **Servers > Web Servers > *web_server_name* > Plug-in Properties > Configuration tab > Plug-in logging**
- Default location:
`<plugin_root>/logs/<web_server_name>/http_plugin.log`
- Set the Log level to **Trace** to trace all the steps in the HTTP request process (caution: this trace produces much log data)

Plug-in logging:

* Log file name
`c:\Program Files\IBM\HTTPServer\Plugins\logs\webserver1\http_plugin.log`

Log level
Error

Trace
Stats
Warn
Error
Debug
Detail

Apply Reset Cancel



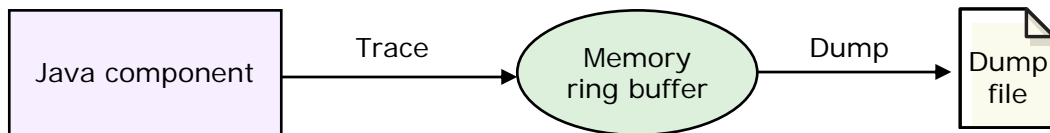
Diagnostic tracing

- Diagnostic tracing can be used to collect detailed processing data for all WebSphere Application Server components
- To take advantage of tracing, you must:
 - Enable tracing of one or more WebSphere components
 - Configure and view trace logs
 - Interpret trace logs and trace messages
- Trace files show the time and sequence of methods that WebSphere Application Server base classes call
 - You can use these files to pinpoint the failure



Using diagnostic tracing

- Tracing can be started:
 - While the server is running, by using Runtime Diagnostic Trace
 - When the server is starting and running, by using Configuration Diagnostic Trace
- Trace output can be directed to:
 - File (default)
 - Memory ring buffer, and dumped after trace stops
- Tracing has a significant affect on performance
 - Enable temporarily for problem determination
 - Tracing to a file is slower than tracing to a memory ring buffer



Enable and configure tracing

- Troubleshooting > Logs and Trace > *server_name* > Diagnostic Trace
- Configure Trace Output
 - None
 - Memory buffer
 - File (default)
- Configure Trace Output Format
 - Basic (IBM Support preference)
 - Advanced
- **Note:** Configure Log Detail Level to get trace output

The screenshot shows the 'Configuration' tab of the WebSphere Configuration Console. The 'Runtime' sub-tab is selected. Under the 'General Properties' section, the 'Trace Output' settings are visible. The 'File' radio button is selected. The 'Maximum Buffer Size' is set to 8 thousand entries. The 'Maximum File Size' is set to 20 MB. The 'Maximum Number of Historical Files' is set to 5. The 'File Name' is set to `${SERVER_LOG_ROOT}/trace.log`. The 'Trace Output Format' is set to 'Basic (Compatible)'. At the bottom, there are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

Configuration Runtime

General Properties

Trace Output

☐ None

☐ Memory Buffer

* Maximum Buffer Size
8 thousand entries

☒ File

* Maximum File Size
20 MB

* Maximum Number of Historical Files
5

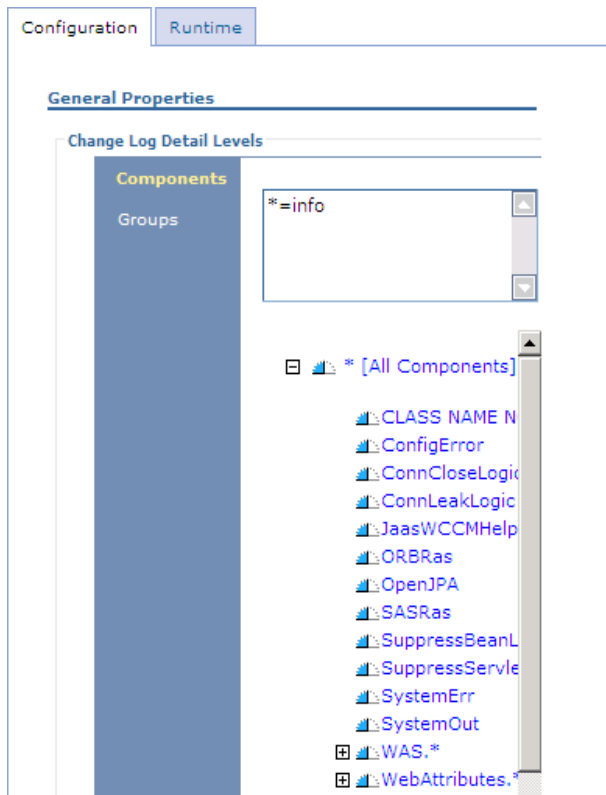
* File Name
`${SERVER_LOG_ROOT}/trace.log`

Trace Output Format
Basic (Compatible)

Apply OK Reset Cancel

Setting the log detail level (1 of 2)

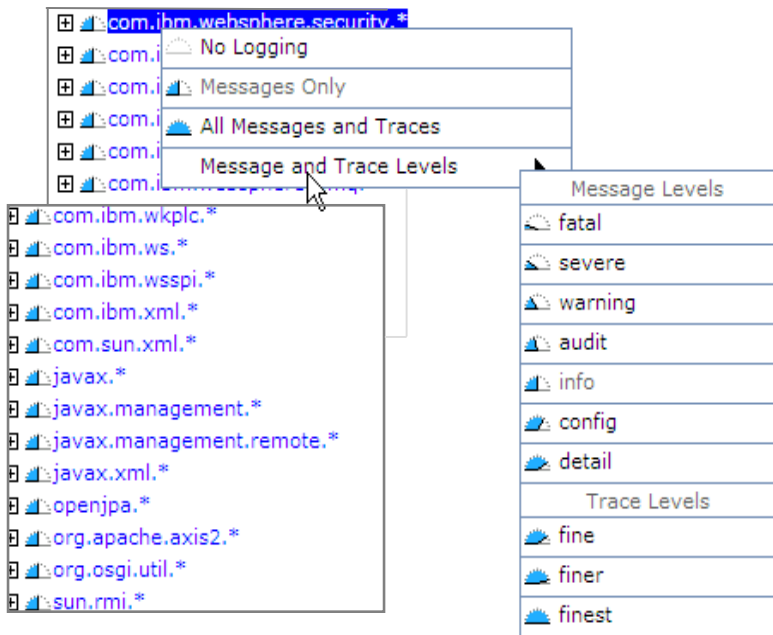
- **Logs and trace > *server_name* > Change Log Detail Level**
- Log detail level affects tracing *and* regular logging
 - Setting levels below **info** reduces the amount of data in logs
 - ***=off** disables logging altogether
- Trace levels (**fine**, **finer**, **finest**) are not displayed in the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





Setting the log detail level (2 of 2)

- Select component
- Select one of
 - No Logging
 - Messages Only
 - All Messages and Traces
- Or select **Message and Trace Levels** and select **Message Levels**



Trace output content and format

- Trace output allows administrators to examine processes in the application server and diagnose various issues
- Two formats can be configured
 - Basic (IBM Support preference)
 - Advanced
- Trace events that are displayed in basic use the following format:
 - `<timestamp><threadId><shortName><eventType>[className]
[methodName]<textmessage> [parameter 1] [parameter 2]`
- Possible values of **eventType** include:
 - > a trace entry of type method entry
 - < a trace entry of type method exit
 - 3 a trace entry of type finest, debug, or dump



Reading a log or trace file (1 of 2)

- Example log record format

- [5/9/11 12:27:56:237 EDT] 00000000 PMIImpl A CWPMI1001I:

PMI is enabled

- Timestamp = [5/9/11
12:27:56:237 EDT]
- Thread ID = 00000000
- Logger = PMIImpl
- Message type = A
- Message code =
CWPMI1001I
- Message =
PMI is enabled

Message type	Description
1,2,3	Trace information: fine, finer, finest
A	Audit
W	Warning
Z	Type was not recognized
E	Error
D	Detail
C	Configuration
F	Fatal (exits process)
I	Information
O	Program output (system.out)
R	Program output (system.err)



Reading a log or trace file (2 of 2)

- Timestamps give good clues:
 - Timestamps are real host time values
 - Good when comparing traces from different processes and correlating events of different servers
- Look for exceptions (search for exception from top of stack trace)
 - Events before the exception are probable causes
 - Events after the exception are recovery attempts
- Often useful to follow a single thread
 - Use the Thread ID to gather related messages

High Performance Extensible Logging (HPEL)

- HPEL can be enabled on any server in the cell
 - Deployment manager
 - Node agent
 - Application server
- Click **Troubleshooting > Logs and trace > server_name**
 - Click **Switch to HPEL mode**

Logging and tracing ?

[Logging and tracing](#) > **server1**

It is recommended that you switch to High Performance Extensible Logging (HPEL) if you have no existing procedures that prevent you from taking advantage of it.

[Switch to HPEL Mode](#) (Advised for most installations)

Use this page to select a system log to configure, or to specify a log detail level for components and groups of components. Use log levels to control which events are processed by Java logging.

General Properties

- [Diagnostic Trace](#)
- [JVM Logs](#)
- [Process Logs](#)
- [IBM Service Logs](#)
- [Change log detail levels](#)
- [NCSA access and HTTP error logging](#)

HPEL logging and tracing configuration

- Use this page to configure HPEL logging and tracing

Logging and tracing ? -

[Logging and tracing](#) > server1

General Properties

[Configure HPEL logging](#)

Directory	/opt/IBM/WebSphere/AppServer/profiles/p1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of logs	50 Megabytes

[Configure HPEL trace](#)

Directory	/opt/IBM/WebSphere/AppServer/profiles/p1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of trace	50 Megabytes

[Configure HPEL text log](#)

Current status:	Enabled
Directory	/opt/IBM/WebSphere/AppServer/profiles/p1/logs/server1
For cleanup, delete records older than	Disabled
For cleanup, maximum size of text log	50 Megabytes

Configure HPEL logging

- Clicking the **Configure HPEL logging** link starts the configuration page
- Changes that are made on the **Configuration** tab require you to restart the server
- Changes that are made on the **runtime** tab take effect immediately

The screenshot shows the 'Logging and tracing' configuration window for 'server1'. The 'Configuration' tab is selected. The 'General Properties' section includes a text field for the directory path set to '\${SERVER_LOG_ROOT}', and two checked checkboxes: 'Enable log record buffering' and 'Start new log file daily at' (set to 12 AM). The 'Log record purging policies' section has a checked checkbox for 'Begin cleanup of oldest records' (set to 'when log size approaches maximum'), a 'Log record age limit' of 48 hours, and a 'Maximum log size' of 20 megabytes. The 'Out of space action' is set to 'Stop logging'. At the bottom are buttons for 'Apply', 'OK', 'Reset', and 'Cancel'.

Logging and tracing

[Logging and tracing](#) > [server1](#) > HPEL Log Configuration

Configuration Runtime

General Properties

* Directory path
\${SERVER_LOG_ROOT}

☒ Enable log record buffering

☒ Start new log file daily at: 12 AM

Log record purging policies

☒ Begin cleanup of oldest records
when log size approaches maximum

Log record age limit
48 Hours old

Maximum log size
20 Megabytes

* Out of space action
Stop logging

Apply OK Reset Cancel

The HPEL Log Viewer

- Click **Troubleshooting > Logs and trace > *server_name* > View HPEL logs and trace**
 - There are numerous filtering options to modify which records are displayed

Logging and tracing

[Logging and tracing](#) > [server1](#) > Log Viewer

Content and Filtering Details

Refresh View Show Only Selected Threads Show All Threads Select Columns ... Export ... Copy

Viewing log records from server instance June 3, 2011 09:55:44

Number of records to show: [First Page](#)

Time Stamp	Thread ID	Logger	Level	Message
6/3/11 09:55:44.896	00000000	com.ibm.ejs.ras.ManagerAdmin	INFO	TRAS0017 : The startup trace state is *info.
6/3/11 09:55:44.940	00000000	com.ibm.ejs.ras.ManagerAdmin	INFO	TRAS0111 : The message IDs that are in use are d
6/3/11 09:55:45.094	00000000	com.ibm.ws.config.ModelMgr	INFO	WSVR0800 : Initializing core configuration models



LogViewer command-line tool (1 of 2)

- Use the **LogViewer** command to query the contents of the High Performance Extensible Logging (HPEL) log and trace repositories
 - `logViewer.sh/bat`
- You can also use the **LogViewer** command to view new log and trace repository entries as the server writes content to them
 - `logViewer.sh/bat -monitor [interval]`



LogViewer command-line tool (2 of 2)

- To get a listing of available **logViewer** options:
 - `logViewer.sh/bat -help`
- The **logViewer** command started from the profile `bin` directory without any options results in a display of all the log and trace messages of a server
- To view the log and trace entries of the most recent server, run:
 - `logViewer.sh/bat -latestInstance`
 - With this option, it is no longer necessary to clear log files before restarting a server

What is Cross Component Trace (XCT)?

- A feature that annotates the logs so that entries that are related to a request are identified as belonging to the same unit of work
- The request might traverse more than one
 - Thread
 - Process
 - Server
- XCT helps identify the root cause of problems across components, which provides the following benefits:
 - Enables administrators and support teams to follow the flow of a request from end-to-end.
 - The request is traced as it traverses thread or process boundaries, or travels between stack products and WebSphere Application Server
 - Helps to resolve questions about which component is responsible for a request that fails

Administering XCT

- A server must have HPEL enabled before XCT can be enabled
- Click **Troubleshooting > Logs and trace > *server_name* > Change log detail levels**
- Check **Enable log and trace correlation**

Correlation

Enable log and trace correlation so entries that are serviced by more than one thread, process, or server will be identified as belonging to the same unit of work.

- ☒ Enable log and trace correlation
- ☒ Include request IDs in log and trace records
- ☐ Include request IDs in log and trace records and create correlation log records
- ☐ Include request IDs in log and trace records, create correlation log records, and capture data snapshots

- Select option for including request IDs, creating correlation logs, capturing data snapshots

XCT request IDs

- XCT request IDs are identifiers added to log and trace records that the server produces
- XCT adds the same request ID to every log or trace record that is a part of the same request, regardless of which thread or JVM produces the record
- When XCT is used with the HPEL log and trace infrastructure, you can view request IDs when logs are output in advanced format
 - `logViewer.sh -minLevel WARNING -format advanced`

```
[Time_stamp] 00000094  W UOW= source=com.ibm.ws.webcontainer.srt  
class=com.ibm.ws.webcontainer.srt.SRTServletResponse method=setIntHeader  
org= prod= component= thread=[WebContainer : 4]  
requestID=[AAAsirk1Njr-AAAAAAAAAA+] appName=[PlantsByWebSphere]
```



Use XCT request ID information to track requests

- Filter your logs by request ID by using the HPEL LogViewer command-line tool

- `logViewer.sh -includeExtensions requestID=AAAsirk1Njr-AAAAAAAAAA+`

```
[Time_stamp] 00000094 XCT          I   BEGIN AAAsirk1Njr-AAAAAAAAAA+
000000000000-cccccccccc2 HTTPCF(InboundRequest
/PlantsByWebSphere/javax.faces.resource/jsf.js.jsf
RemoteAddress(127.0.0.1) RequestContext(-957274864))

[Time_stamp] 00000094 srt          W
com.ibm.ws.webcontainer.srt.SRTServletResponse setIntHeader SRVE8094W:
WARNING: Cannot set header. Response already committed.

[Time_stamp] 00000094 XCT          I   END   AAAsirk1Njr-AAAAAAAAAA+
000000000000-cccccccccc2 HTTPCF(Request AsyncWrite RequestContext(-
957274864))
```



Gathering diagnostic data





Gathering JVM diagnostic data

- A wealth of valuable JVM diagnostic data can be gathered by doing the following tasks:
 - Enable logging of verbose garbage collection data for servers
 - Generate a Java thread dump
 - Generate a Java heap dump
 - Generate a system core dump
 - Configure hung thread detection
 - Enable tracing of connection leaks

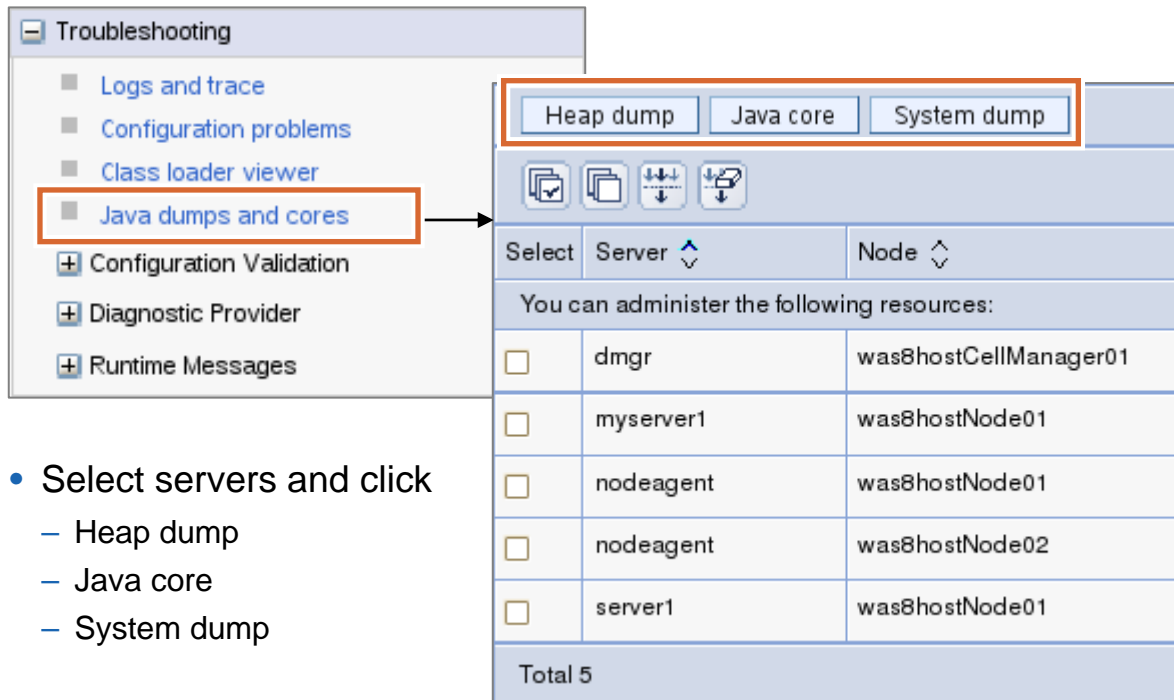


Enable verbose garbage collection

- Verbose GC is an option that the JVM run time provides
- Provides a garbage collection log
 - Interval between collections
 - Duration of collection
 - Compaction required
 - Memory size, memory that was freed, memory available
- Turns on verbose GC for each server through the administrative console
 - **Servers > *server_name* > Process Definition > Java Virtual Machine**
 - Select “Verbose Garbage Collection” check box
 - Save and distribute
 - Restart the server or servers
- Usually writes to `native_stderr.log` file
 - Varies depending on platform and WebSphere version
 - Some load on system resources because of disk I/O, but minimal unless thrashing

Java memory dumps and cores

- New feature in the Troubleshooting section is **Java dumps and cores**



The screenshot displays the WebSphere Troubleshooting console. On the left, a sidebar lists several options: 'Logs and trace', 'Configuration problems', 'Class loader viewer', 'Java dumps and cores' (highlighted with a red box), 'Configuration Validation', 'Diagnostic Provider', and 'Runtime Messages'. An arrow points from the 'Java dumps and cores' option to the main content area. The main area features three buttons at the top: 'Heap dump', 'Java core', and 'System dump', which are also highlighted with a red box. Below these buttons are four icons representing document operations. Further down, there are 'Select', 'Server' (with a dropdown arrow), and 'Node' (with a dropdown arrow) controls. A text box states 'You can administer the following resources:'. Below this is a table with five rows, each containing a checkbox, a server name, and a node name. The table is summarized with 'Total 5' at the bottom.

	Server	Node
<input type="checkbox"/>	dmgr	was8hostCellManager01
<input type="checkbox"/>	myserver1	was8hostNode01
<input type="checkbox"/>	nodeagent	was8hostNode01
<input type="checkbox"/>	nodeagent	was8hostNode02
<input type="checkbox"/>	server1	was8hostNode01

Total 5

- Select servers and click
 - Heap dump
 - Java core
 - System dump



Generating a JVM thread dump (javacore)

- What is a javacore?
 - Small diagnostic text file that the JVM produces
 - Contains much vital information about the running JVM process
 - Provides a snapshot of all the running threads, their stack traces, and the monitors (locks) held by the threads
 - Useful for detecting hang or deadlock conditions
- How to manually trigger a thread dump
 - Warning: triggering a thread dump can terminate the server process
 - Use operating system facilities:
`kill -3 <JVM PID>` (UNIX or Linux)
 - Explicitly tell WebSphere to generate a thread dump
 - Issue wsadmin Jacl or Jython commands

Generating a JVM heap dump

- JVM heap dump is created in the `<profile_root>` directory when an `OutOfMemoryError` exception is thrown
 - Heap dump is in phd (portable heap dump) format by default
 - Different locations and formats can be configured by using command-line arguments
- Use the `-Xdump` generic JVM arguments to configure heap dumps
- For example:
 - `Xdump:heap:file=/dumps/heapdump.%Y%m%d.%H%M%S.%pid.%seq.phd` specifies the location and format of the heap dump file name
 - `Xdump:heap, opts=PHD+CLASSIC` enables heap dump creation and creates the file in both binary and text format
 - `Xdump:heap:none` disables heap dump creation
- Can also trigger heap dumps by using wsadmin Jython commands
 - `AdminControl.invoke(AdminControl.completeObjectName("type=JVM, process=<server_name>,*"), "generateHeapDump")`

Hung thread detection

- Hung threads can be hard to diagnose
 - They are often not noticed until many threads are hung, resulting in a performance problem
- Application threads can hang for a number of reasons, which include:
 - Endless loops
 - Deadlocks
 - Inaccessible resources
- The **ThreadMonitor** component monitors the web container, ORB, and Async bean thread pools for hung threads
- Example thread monitor message

```
[8/1/11 10:51:01:955 EDT] 00000021 ThreadMonitor W
WSVR0605W: Thread "WebContainer : 0" (00000020) has been
active for 65103 milliseconds and may be hung. There is/are
1 thread(s) in total in the server that may be hung.
```



Connection leak diagnostic messages

- Poorly written applications often do not properly release database connections
 - Forget to call `connection.close()`
 - Most often in the exception case
 - Connections must be closed in a `finally{}` block
- Orphaned connections return to the pool only after timeout
 - Can cause a backup of new connections that are waiting for old connections to timeout
 - New connections that wait too long throw a `connectionWaitTimeoutException`
- Connection manager is instrumented to print stack traces when a `connectionWaitTimeoutException` occurs
 - Enable by using trace string `WAS.j2c=finest`



Checking version levels and applying APARs

- The **versionInfo** command generates a report from data that is 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 that customers report
- Download suggested fixes, and fixes by version from the WebSphere Support page
- IBM Installation Manager: new standard tool for installing fixes



Problem determination tools



Problem determination tool availability

- Some tools are integrated into the administrative console
 - Tivoli Performance Viewer
 - Configuration validation
 - Class loader viewer
 - Others
- Some tools are separate programs that are shipped with WebSphere Application Server itself, or shipped as separate products in their own right
 - dumpNameSpace
 - versionInfo
 - Rational Application Developer
- IBM Support Assistant:
 - Serves as a central point from which many tools can be found and even run directly inside IBM Support Assistant



Problem determination tools: Administrative console

- Tivoli Performance Viewer and Performance Advisor
 - Captures and analyzes statistical performance data from a running WebSphere Application Server system
- Configuration validation
 - Checks for errors in a WebSphere Application Server configuration
- Class Loader Viewer
 - Provides a hierarchical view of class loaders that applications use
 - Helps diagnose problems that are related to loading of classes in applications
- Diagnostic Provider
 - Tool for viewing configuration and the current state of individual application server components



Dumping the JNDI namespace

- The `dumpNameSpace` utility shows JNDI directory content
- Useful to ensure correct association of named objects:
 - JDBC resources
 - EJBs
 - JMS resources
 - Other resources
- Syntax and some of the options:

```
<was_root>\bin\dumpNameSpace  
  [-host bootstrap_host_name (defaults to localhost)]  
  [-port bootstrap_port_number (defaults to 2809)]  
  [-startAt subcontext/in/the/tree]
```

- Output can be redirected to a file and inspected

IBM Support Assistant





What is the IBM Support Assistant?

- The IBM Support Assistant workbench is a free, stand-alone application that is installed on any workstation
- It can also be enhanced by installing plug-in modules for the IBM products you use
 - The IBM Support Assistant workbench is supported on Windows and Linux
 - The IBM Support Assistant Agent manager and Assistant agent must be configured to support other platforms such as AIX and Solaris
- Benefits of IBM Support Assistant include:
 - Saves time in searching product, support, and educational resources
 - Helps with opening a problem management report (PMR)
 - Allows for easy location and installation of useful product support tools by using a support tool framework
 - Easily downloaded from
<http://www.ibm.com/software/support/isa/>


IBM Support Assistant workbench home page

Support Assistant

Launch Activity Home


Welcome

Welcome to IBM Support Assistant




Find Information

Easily find the information you need including product specific information and search capabilities.




Analyze Problem

Diagnose and analyze problems through serviceability tools, collection of diagnostic artifacts, and guidance through problem determination.




Collect and Send Data

Collect problem determination files using automated data collection. Use these files for self-help problem determination, or use the Service Request function to send the files with a service request to IBM.



First Steps

Take your first steps



Tutorials

Go through tutorials

Latest News

- Update! Memory Analyzer V1.1.0
- New! Multicore SDK tools
- Update! Get IBM Support Assistant V4.1.2
- Update! Health Center 1.3
- More News...

IBM



IBM Support Assistant: Search component

- Find **Information > Search Information**
- Allows searching multiple information locations concurrently, including:
 - Technical notes from the IBM software support website
 - Topics from IBM developerWorks
 - Postings from IBM newsgroups and forums
 - Google web search
 - Online information centers
- You can also narrow your search by selecting specific IBM software products and versions



IBM Support Assistant: Tools component

- Select **Analyze Problem > Tools**
 - Provides support tools to identify and troubleshoot problems
- Configuration tools
 - **Port Scanning Tool**: scan system for active and configured ports
 - **Visual Configuration Explorer**: visualize, explore, and analyze configuration information from diverse sources
 - Others
- JVM-based tools
 - **Thread and Monitor Dump Analyzer**: analyze javacore files
 - **Garbage Collector and Memory Visualizer**: analyze verbose GC logs
 - **Memory Analyzer**: analyze heap dumps for leak suspects
 - **Java Health Center**: monitor active JVMs with minimal effect on performance
 - Others

IBM Support Assistant: Collect Data component

- Select **Analyze Problem > Collect Data**
 - Provides data collectors for many different problems
 - Interactive collection process
- System collectors
 - Collect system data information
 - Collect network information
 - Collect software inventory
 - Others
- WebSphere Application Server collectors
 - Security
 - Connectors
 - Containers
 - JDK
 - Runtime
 - Others



IBM Support Assistant: Collect and send data

- Select collect data
 - Local collection with IBM Support Assistant Workbench
 - Remote collection with IBM Support Assistant Lite
 - Remote collection with agents
- Select service request
 - Open a new service request
 - Review or update service requests

IBM Support Assistant 5.0 Beta

- The new IBM Support Assistant 5.0 Beta provides a web browser-based version of the problem determination platform

The screenshot shows the IBM Support Assistant web application running in a Mozilla Firefox browser. The address bar displays the URL: `http://localhost:10911/ja5/#view=tools&id=0001&toolId=com.ibm.java.diagnostic`. The browser tabs include "IBM Support Assistant", "IBM Thread and Monitor Dump Analyze...", and "Eclipse Memory Analyzer".

The application interface features a top navigation bar with "Cases", "Files", "Tools", "Reports", and "Data Collector". Below this is a search bar labeled "Search Tool Help". The main content area is divided into two panels. The left panel, titled "Enter keyword", contains a list of tools with checkboxes and icons. The tool "Garbage Collection and Memory Visualizer (GCMV) [Desktop]" is selected and highlighted with a dashed blue border. Other tools listed include "Garbage Collection and Memory Visualizer (GCMV) [Report]", "Health Center", "Memory Analyzer [Desktop]", and "Memory Analyzer [Report]".

The right panel displays the details for the selected tool, "Garbage Collection and Memory Visualizer (GCMV) [Desktop]". It includes a "Launch" button, a "Tool Help" link, and a "Description" section. The description states: "IBM Monitoring and Diagnostic Tools for Java(TM) - Garbage Collection and Memory Visualizer (GCMV). This tool is a verbose GC data visualizer. It parses and plots various log types including verbose GC logs, -Xtrac output and".

At the bottom of the interface, the build ID is shown as "5.0.0.8_B_20120528-2008", and the copyright notice reads "© Copyright IBM Corp. 2011, 2012. All rights reserved | [Terms of use](#)". A status bar at the very bottom indicates "Transferring data from localhost...".



IBM Support Assistant 5.0 Beta: New features

- New and improved features:
 - **Reports view:** Provide a quick, concise view into all reports that are generated from the execution of a report generator analysis tool
 - **Information center:** Expanded documentation for IBM Support Assistant and Tools and now built into the application
 - **Desktop tools:** Desktop tools can be started by using the IBM Support Assistant browser UI (Note: requires a Java plug-in in your browser)
 - Web-based tools: Rich UI tools run in your browser while the heavier processing of diagnostic files is offloaded to the IBM Support Assistant server
- Multiple installation and deployment options:
 - EAR deployment: New easy deployment of an EAR file to your existing WebSphere Application Server
 - Extract and go: Simple installation by extracting a download package and starting IBM Support Assistant. This new embedded server (Liberty profile) has a fast startup time, fast page load, and small footprint



Using the IBM Support Assistant Data Collector (1 of 2)

- The IBM Support Assistant Data Collector for WebSphere Application Server is a tool for gathering diagnostic data from an application server
 - It replaces the collector tool, which is deprecated
- Also provides symptom analysis support for the various categories of problems
- The tool runs in console mode by starting the launch script from the command line
- In a Windows environment, run the
`<profile_root>/bin/isadc.bat` command
- In a Linux, AIX, HP-UX, Solaris, IBM i, or zOS environment, run the
`<profile_root>/bin/isadc.sh` command



Using the IBM Support Assistant Data Collector (2 of 2)

```
Terminal
File Edit View Terminal Tabs Help
was85host:/opt/IBM/WebSphere/AppServer/profiles/profile1/bin # ./isadc.sh
Starting IBM Support Assistant Data Collector in console mode...

ApplicationServer: 2.0.2.20120316
IBM Support Assistant Data Collector:
2.0.1.GA20120316-1743
Common Inventory Sub Agent: 6.3.0.20120123

Create the collection zip file at /root/localhostNode01Cell-
was85host01Node01-WAS-ISADC.zip?
[1] Yes
[2] No
> 1

Enter the number of the IBM Support Assistant Data Collector option you
want to execute.
[1] Default Collection           [ distributed, IBM i, z/OS ]
[2] General...                  [ distributed, IBM i ]
[3] Administration...          [ distributed, IBM i ]
[4] Security...                 [ distributed, IBM i ]
[5] Connectors...              [ distributed, IBM i ]
[6] Containers...              [ distributed, IBM i ]
[7] JDK...                     [ distributed ]
[8] Runtime...                 [ distributed, IBM i ]
[9] HTTP...                    [ distributed, IBM i ]
[10] Service Oriented Architecture... [ distributed, IBM i ]
[11] Intelligent Management Pack... [ distributed, IBM i ]
[12] Quit
```

Unit summary

Having completed this unit, you should be able to:

- Describe a basic approach for problem determination
- List resources for completing a problem investigation
- Locate relevant log files
- Examine log activity
- Enable tracing on specific components
- Enable high performance extensible logging (HPEL) and use the Log Viewer
- Locate other troubleshooting tools
- Use the IBM Support Assistant



Checkpoint questions

1. What are the two approaches for problem determination?
2. In basic mode logging, what are the two types of JVM log files, and what data do they contain?
3. What are the three levels of diagnostic trace detail?
4. What are the three major features of the IBM Support Assistant that are accessible from the Welcome screen?
5. What are some JVM-related problems that a server can experience?

Checkpoint answers

1. What are the two approaches for problem determination?
 - Analysis and isolation
2. In basic mode logging, what are the two types of JVM log files, and what data do they contain?
 - The `SystemOut.log` file contains messages from different server components. Mostly informational, these messages log events that occur during the lifetime of the JVM. The `SystemErr.log` file contains any exceptions and stack traces that server components throw.
3. What are the three levels of diagnostic trace detail?
 - Fine, finer, and finest
4. What are the three major features of the IBM Support Assistant that are accessible from the Welcome screen?
 - Find Information, Analyze Problem, Collect, and Send Data
5. What are some JVM-related problems that a server can experience?
 - Hung threads, OutOfMemory conditions, connect leaks



Exercise 7



Problem determination



Exercise objectives

After completing this exercise, you should be able to:

- Use the administrative console to configure and view log data
- Enable a server to use HPEL
- Enable tracing on application server components
- Use the HPEL Log Viewer to examine log and trace data
- Enable verbose garbage collection for an application server
- Enable memory leak detection for an application server
- Describe how IBM Support Assistant tools can be used to analyze JVM memory dumps