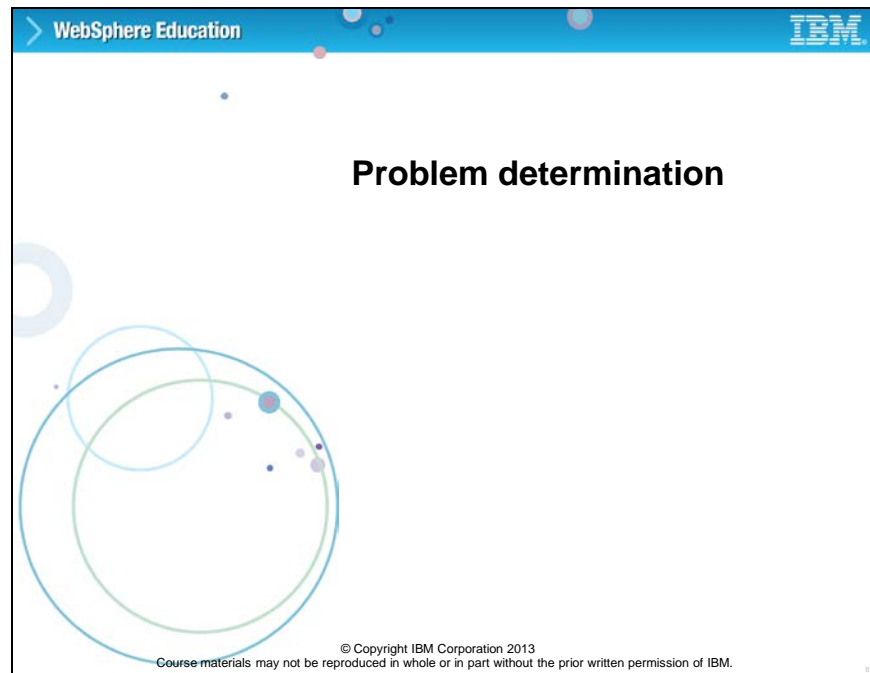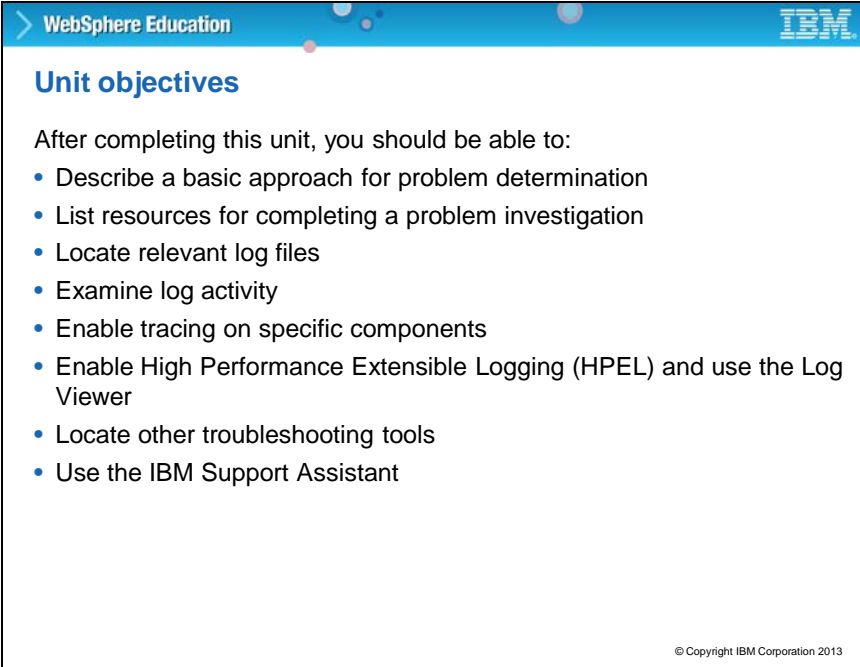Slide 1

Unit 11: Problem determination

This unit describes the tools available to the administrator for investigating problems and gathering diagnostic data.

1

Slide 2

> **WebSphere Education** IBM
>
> ## 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
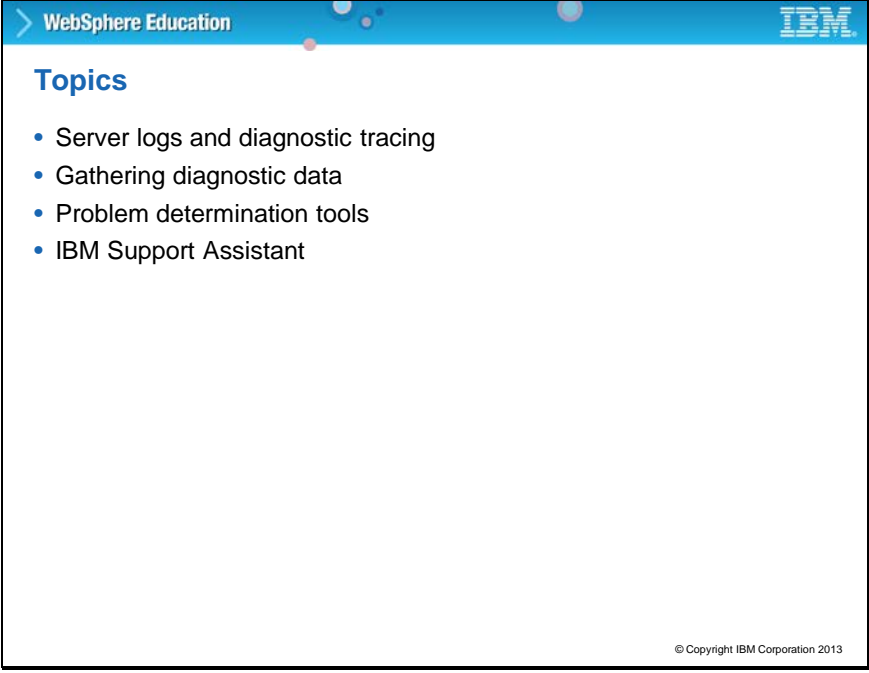> - Use the IBM Support Assistant
>
> © Copyright IBM Corporation 2013

Title: 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

Slide 3

Topics:

The unit covers the following topics:

- Server logs and diagnostic tracing

- Gathering diagnostic data

- Problem determination tools

- IBM Support Assistant

Slide 4

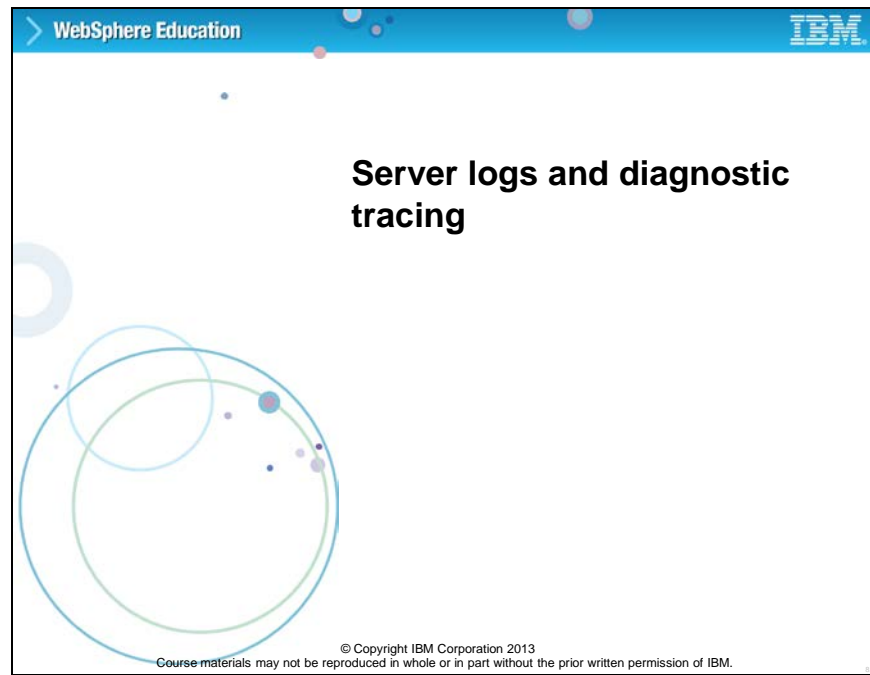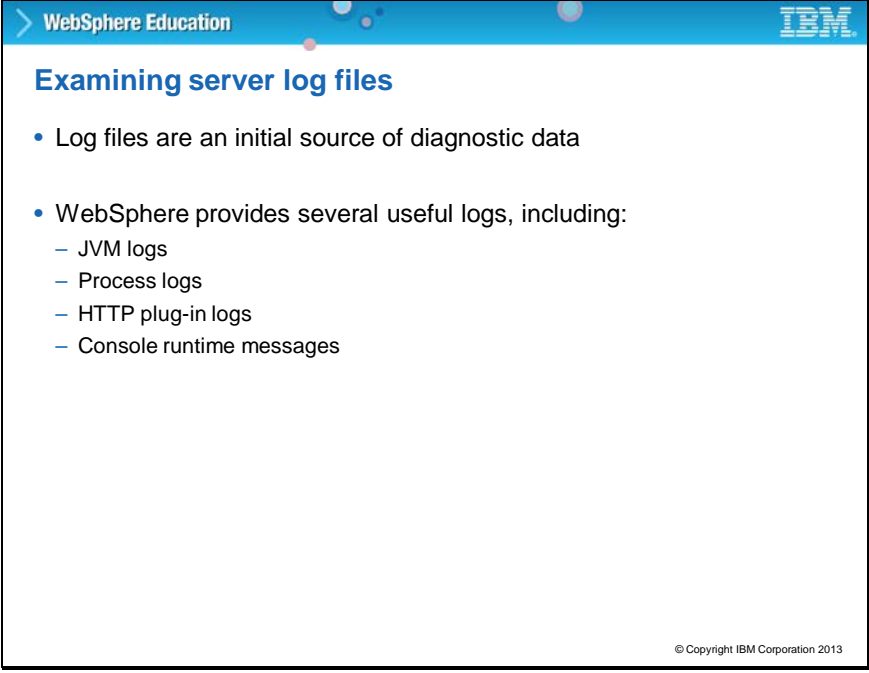**Topic: Server logs and diagnostic tracing**

This unit describes the different server log files and diagnostic tracing facilities.

Slide 5

Title: Examining server log files

WebSphere provides several useful logs and messaging facilities, including JVM logs, HTTP plug-in logs, and console runtime messages.
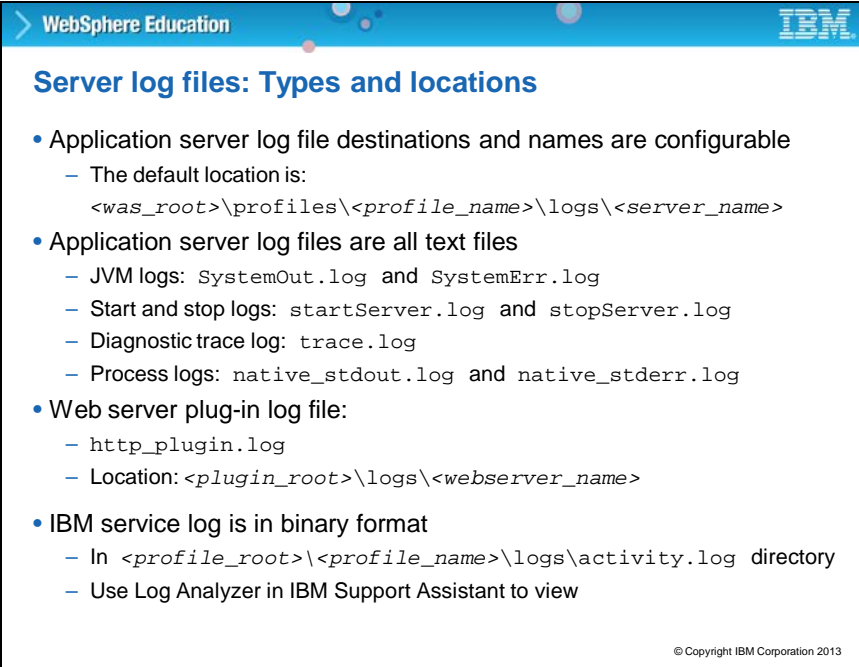
Slide 6

Title: WebSphere Application Server logs

The JVM logs are created by redirecting the system.out and system.err streams of the Java virtual machine to independent log files. There is one set of JVM logs for each application server and all of its applications. They are located, by default, in the directory shown.

Applications and other code can write to these streams by using the print() and println() methods of those streams. In a network deployment configuration, JVM logs are also created for the deployment manager and each node agent.

The process logs include two output streams, standard out and standard error, which is accessible to local code that is running in the process. There is one set of logs for each application server. The IBM service log contains the WebSphere Application Server messages that are written to the system.out stream. It also contains some special messages that contain extended service information. There is a single log file for each WebSphere profile named activity.log.

Slide 7

Title: Server log files: Types and locations

The names and destinations of the log files can be configured. The default names and locations are listed here. The application server logs are in the **WebSphere root > profiles >** *profile name* **> logs** directory, by default. The web server plug-in log file is in the **plug-in root > logs >** *web server name* directory, by default.

Slide 8

Title: Configuring JVM logs

This screen shows the configuration tab in the administrative console where you can configure the location and names of the JVM log files.

For both log files, systemOut and systemErr, you can specify the path to their location, file formatting (basic or advanced), and log file rotation (by file size or time interval). The maximum number of historical files to store on the file system can also be configured.

Slide 9

Title: Viewing runtime messages in the console (1 of 2)

You can view runtime messages in the console by selecting **Troubleshooting > Runtime Messages.** The system displays runtime messages that are grouped according to severity levels: error, warning, and information. From the drop-down menu, select **Info** to enable all runtime events.

When viewing runtime messages, first select the **Error**, **Warning**, or **Info** category link (a count of zero means that nothing is available). Then, the details for the selected category are shown. Selecting one of these links gives you detailed information (see next slide).

You can have several pages of messages; the button on the bottom of the page allows you to view and read all of them.

Slide 10

Title: Viewing runtime messages in the console (2 of 2)

Click a message in the runtime messages view to see more details about that message. The runtime events details shown here include: the message code and text, a brief explanation of the event, and other information.

Most runtime messages are designed with improved message text. A message code, for example SRVE0255E, can be used to look up the message in the information center. Information is shown on the detail screen for the event, and sometimes a user action is provided to resolve the problem.

Slide 11

Title: HTTP plug-in logs and tracing

The web server plug-in log file can be configured from the administrative console by selecting

Servers > Web Servers > web server name > Plug-in Properties > Configuration tab > Plug-in logging.

The default location of the web server plug-in log file is in the **plug-in root > logs > web server name** directory.

By setting the log level to trace, all steps in an HTTP request are recorded in the log file. This setting can produce a large amount of log data.

Slide 12

Title: Diagnostic tracing

Diagnostic tracing provides another level of log data and can be used to collect detailed processing information for WebSphere Application Server components. Tracing is not enabled by default, but can be enabled and configured in the administrative console. WebSphere Application Server base classes call trace data to capture the time and sequence of methods. You can use this information to troubleshoot problems with the application server.

Slide 13

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

© Copyright IBM Corporation 2013

Title: Using diagnostic tracing

Tracing can be enabled while the server is running by using the Runtime Diagnostic trace settings in the administrative console. Tracing can also be configured to start when the server is started.

Trace output allows administrators to examine processes in the application server and diagnose various issues. On an application server, 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. On an application client or stand-alone process, trace output can be directed either to a file or to the process console window. In all cases, trace output is generated as plain text in either basic or advanced formats, or in binary log analyzer format as the user chooses. The basic and advanced formats for trace output are similar to the basic and advanced formats that are available for the JVM message logs.

Slide 14

Title: Enable and configure tracing

The Diagnostic Trace Service box looks almost the same as it did in previous versions. The configuration and runtime tabs behave as they always have. The configuration changes affect the configuration repository and take effect at the next startup. The run time takes effect immediately but is only optionally persisted to the server configuration.

Trace strings must be configured on a separate panel (Log Detail Level).

View and dump options are available in the runtime tab of diagnostic trace. The Log Analyzer can be used to analyze trace output, but you can use your favorite editor. Before you can view or dump a trace, you must specify the log detail level.

14

Slide 15

Title: Setting the log detail level (1 of 2)

You can set the log detail level in the administrative console by selecting **Logs and Trace > server name > Change Log Detail Level**.

By setting the log detail level to info, you can reduce the amount of output in the logs.

By setting the log detail level above info, you increase the amount of output in the logs and trace data.

Trace levels are fine, finer, and finest. You can specify the log detail level by entering a trace string, or by selecting components from the graphical menu. The default trace string is *= info.

As you select components from the menu, they are added to the trace string. Applications can be instrumented for tracing, and the data they produce is included in the trace output.

Slide 16

Title: Setting the log detail level (2 of 2)

This graphic shows how to set the log detail level from the graphical menu. You can right-click the components and select:

- No Logging
- Messages only
- All Messages and Traces

From the **Message and Trace Levels** menu, you can select a message level of fatal, severe, warning, audit, info, config, or detail. You can select a trace level of fine, finer, or finest.

Slide 17

> **WebSphere Education**   **IBM**

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

© Copyright IBM Corporation 2013

Title: Trace output content and format

Trace output includes events that administrators can analyze to diagnose problems. The basic format for trace messages is shown here. A basic trace message includes a time stamp, thread ID, short name, event type, class name, message name, message text, and one or two parameters. The symbols that are shown here represent an event type. A greater than (>) symbol represents a method entry. The less than symbol (<) represents a method exit. The number three indicates an entry of type finest, debug, or dump.

Slide 18

WebSphere Education   IBM.

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
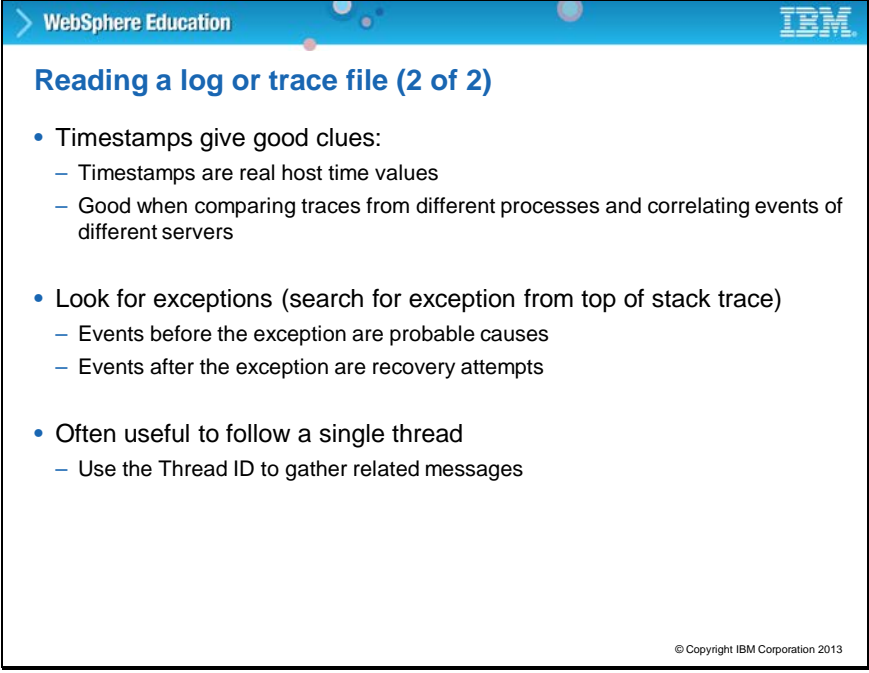 - Message code = CWPMI1001I
  - Message = PMI is enabled

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)

© Copyright IBM Corporation 2013

Title: Reading a log or trace file (1 of 2)

Interpreting trace data takes some practice, but after you get some experience, you learn to recognize familiar messages and patterns. A log record consists of a time stamp, thread ID, name of the component that logged the message, message type, message code, and the text of the message.

The chart that is shown here lists the codes for each message type and their meaning. Some of the message type codes are simple: A stands for audit and W for warning..

Slide 19

Title: Reading a log or trace file (2 of 2)

When reading log or trace data, it is helpful to take note of the time stamps. They can provide clues as to when a problem occurred and what events led up to it. Look for exception messages in the data. Events that occur before the exceptions are probably causes. Events that occur after the exception are recovery attempts. The thread ID indicates messages that are related to a specific thread.

While it is possible to read logs and trace files by using a text editor; it is suggested that you use a tool such as Log Analyzer, which is available in the IBM Support Assistant.

Slide 20

Title: High Performance Extensible Logging (HPEL)

High Performance Extensible Logging, or HPEL, was a new feature in version 8.0. It provides an alternative way of collecting and viewing log data. It replaces the activity.log file. HPEL can be enabled on all servers in the cell, including the deployment manager, node agents, and stand-alone application servers.

To enable HPEL, in the administrative console, select **Troubleshooting > Logs and trace > server_name** and click "Switch to HPEL mode".

After the log level is switched to HPEL mode for a server, there is a new list of links in the **General Properties** section. One new link is **Change log and trace mode**, which allows you to switch back to basic logging.

Slide 21

Title: HPEL logging and tracing configuration

This screen capture shows the default configuration for HPEL logging and tracing. To modify the configuration, click any of the links on this page.

After HPEL is enabled for a server, it is suggested that you disable the HPEL text logs. Disabling text logs improves server performance.

Slide 22

Title: Configure HPEL logging

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

Slide 23

Title: The HPEL Log Viewer

Since HPEL data is stored in binary format, you must view HPEL logs by using a log viewer. In the administrative console, select **Troubleshooting > Logs and trace > server name > View HPEL logs and trace** to view HPEL data. The console provides many options for sorting and filtering the data.

The log view section lists the records. Use the **First Page**, **Previous Page**, **Next Page**, and **Last Page** buttons to move through the list of records. Optionally, you can specify filter criteria in the Content and Filtering Details section to limit the rows that are shown. Records are always listed in the order that the server recorded them. By default, the log view has five columns.

Time Stamp: The time when the event was recorded.

Thread ID: The identity of the thread that recorded the event in hexadecimal notation.

Logger: The logger that recorded the event.

Level: The type of event that was recorded.

Message: The message from the recorded event. If the message has a message ID, the message ID is underlined. Click the message ID to get an explanation and suggested user action for the message.

Slide 24

Title: LogViewer command-line tool (1 of 2)

You can also query the contents of HPEL log and trace data by using the logViewer command that is shown here. By using the monitor option with the command, you can view new log and trace messages as they are written, similar to the way the tail function works in Linux and UNIX.

Slide 25

Title: LogViewer command-line tool (2 of 2)

To get a listing of available LogViewer options, use the -help option.

LogViewer can be started from the bin directory of a profile without any options, and a listing of all the entries of a server is shown. If there are multiple servers in a profile, the user is first prompted to select the server for viewing.

If you start the log viewer from the profile bin directory of a server without specifying any options, it displays all of the log and trace messages for that server. Use the latestInstance option to view the log and trace entries for the most recent server runtime log files. With this option, it is no longer necessity to clear old log files before restarting the server.

Slide 26

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

© Copyright IBM Corporation 2013

Title: What is Cross Component Trace (XCT)?

Depending on the nature of your applications, multiple threads within an application server might be used to handle requests, such as HTTP requests or JMS requests. More than one application server handles some requests, such as when one application server makes a request to another application server for a web service.

Applications that are built by using distributed architectures, such as service-oriented architecture, can benefit from XCT, since XCT helps facilitate problem determination across multiple services on different systems.

Slide 27

Title: Administering XCT

Enable XCT to include request IDs in log and trace files when you want to see which log and trace entries, in all threads and application server processes, are related to the same request. Request IDs are only recorded when using HPEL log and trace mode and can be seen or used for filtering when using the logViewer command.

Select the option "Enable XCT to create correlation log records" when you want to log how requests branch between threads and processes, and see extra information about each request. Enabling XCT to create correlation log records might significantly affect performance on your system, so is best suited for test and development environments.

Slide 28

Title: XCT request IDs

You can use XCT to augment your log and trace files with correlation information. This correlation information clarifies which threads and which application server processes participated in the handling of each request.

Slide 29

Title: Use XCT request ID information to track requests

XCT log records are typically added to the logs to:

- Demarcate the beginning and ending of work for a particular request on a particular thread.
- Demarcate when work is transferred to another thread or process, or to indicate when work returned from another thread or process.
- Demarcate when work moves from major component to major component, even if work continues on the same thread; for example to show transfer of control from application server code to application code.

The HPEL logViewer tool is able to filter log and trace records by request ID.
Tools, such as the XCT Log Viewer, can also take advantage of XCT log records or XCT request IDs, or both, when rendering log and trace content. The XCT Log Viewer is available as a tool add-on for the IBM Support Assistant.
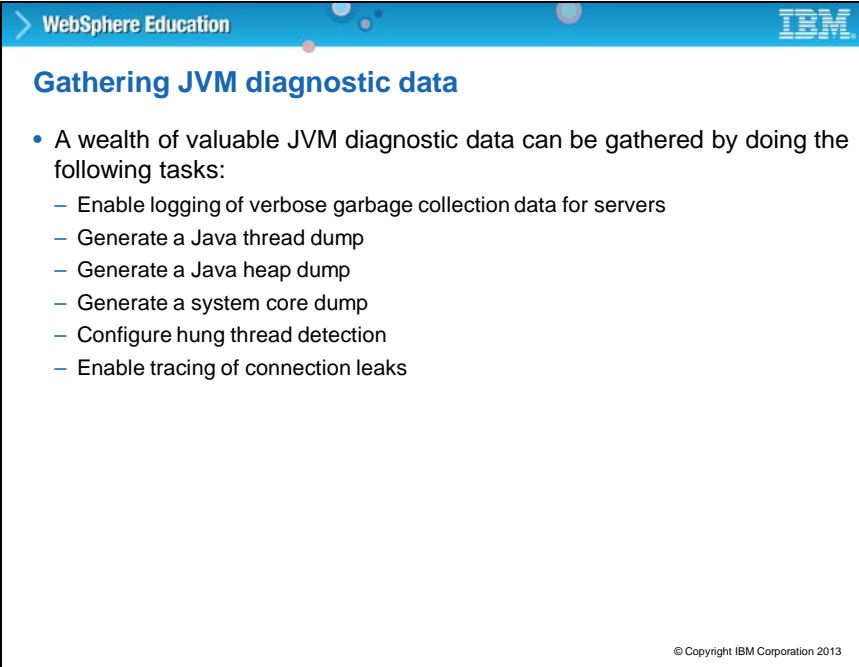
Slide 30
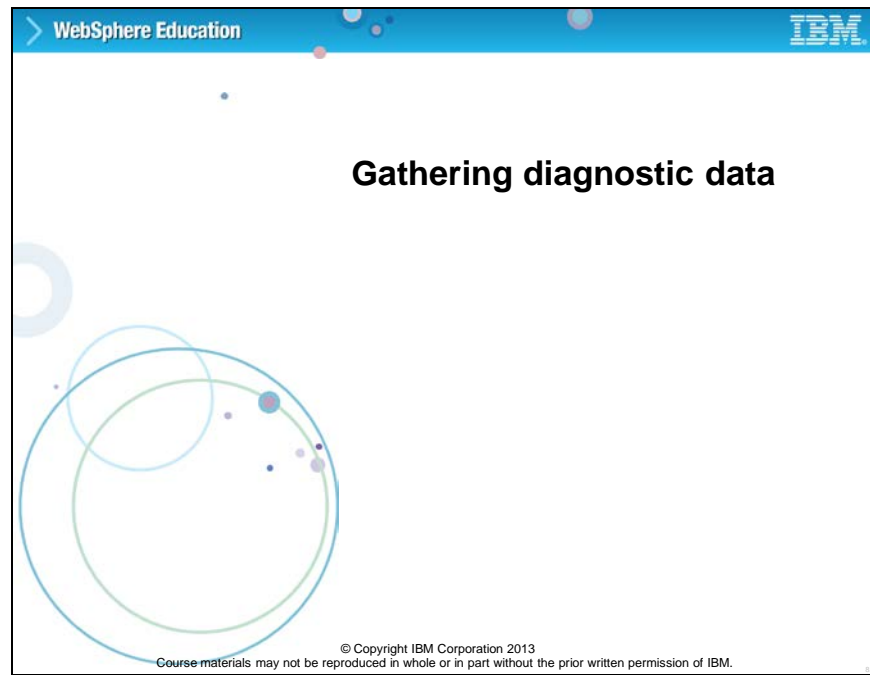
Topic: Gathering diagnostic data

This topic describes how to gather diagnostic data.

Slide 31

Title: Gathering JVM diagnostic data

When investigating a problem related to a JVM, you can gather data by using these techniques:

- Enable verbose garbage collection

- Generate a Java thread dump

- Generate a Java heap dump

- Generate a system dump

- Configure hung thread detection

- Enable diagnostic capability to detect connection leaks

Slide 32

Title: Enable verbose garbage collection

You can enable garbage collection (GC) for each server by using the administrative console. Select **Servers > server name > Process Definition > Java virtual machine** and select the check box for "verbose garbage collection". This action requires a server restart.

The garbage collection log records the interval between collections, duration of collection, compaction requirements, memory size, memory that was freed, and memory available. The cost of enabling verbose garbage collection is minimal. The benefits of having it on the first time something happens are considerable because then there is no need to reproduce the problem. It is also a good practice to monitor the verbose GC regularly as a way to determine the health of the system.

As in previous versions, verbose GC is not enabled by default. When you enable verbose GC for a server in V8.5, the default garbage collection policy is generational-concurrent (gencon). The data is written to the native_stderr.log or native_stdout.log depending on the operating system of the server.

It is often suggested that you have verbose GC enabled permanently in production. The cost on a reasonably well-tuned JVM is small. The benefits of having it on the first time something happens are considerable (no need to reproduce the problem a second time after enabling). It is also good to keep an eye on the verbose GC regularly as a way to monitor the health of the system, even when nothing bad is noticed.

Slide 33

Title: Java memory dumps and cores
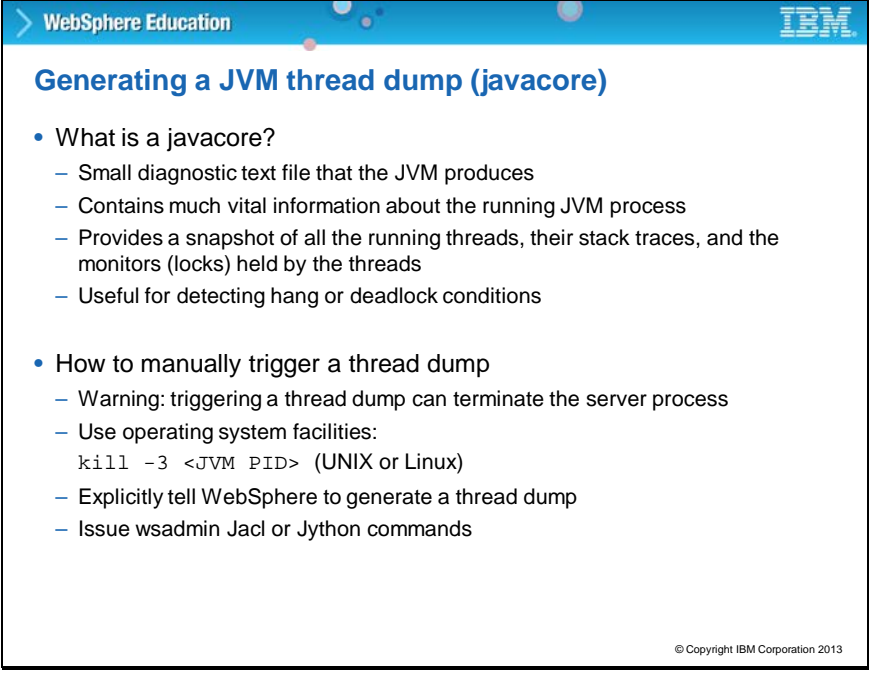
Java dumps and cores is a new feature in version 8. Clicking the **Java dumps and cores** link opens the panel that is shown. Use this panel to generate heap dumps, Java cores, or system dumps for a running process. Select the server and click the appropriate button. The files that result from these operations are placed on the local file system in the profile root directory, by default.

A **heap dump** is a snapshot of JVM memory. It shows live objects in the memory and references between them. Use the heap dump to investigate how applications use memory and to troubleshoot out of memory exceptions.

Use the **Java core** button to generate a thread dump. A thread dump is used to investigate why a server is hanging or investigate messages in the logs that indicate a thread did not complete its work in the expected amount of time.

Use the **System dump** button to generate system dumps of the server process. These dumps can be large, but no other dumps might be available when a server crashes.

33

Slide 34

IBM.

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

© Copyright IBM Corporation 2013

Title: Generating a JVM thread dump (javacore)

A javacore is a small diagnostic text file that the JVM produces. It contains vital information about JVM processes and provides a snapshot of all the threads, their stack traces, and locks held by the threads. A javacore can be used to detect hung threads or deadlock condition.

Thread dumps can also help in detecting performance problems. One method is to take at least three snapshots about 2-3 minutes apart, and then analyze the javacore files to see what different threads are doing in each snapshot.

You can trigger a thread dump by using the operating system command that is shown here, but be aware that triggering a thread dump can crash the server. You can also use wsadmin commands to generate a thread dump.

Slide 35

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")`

© Copyright IBM Corporation 2013
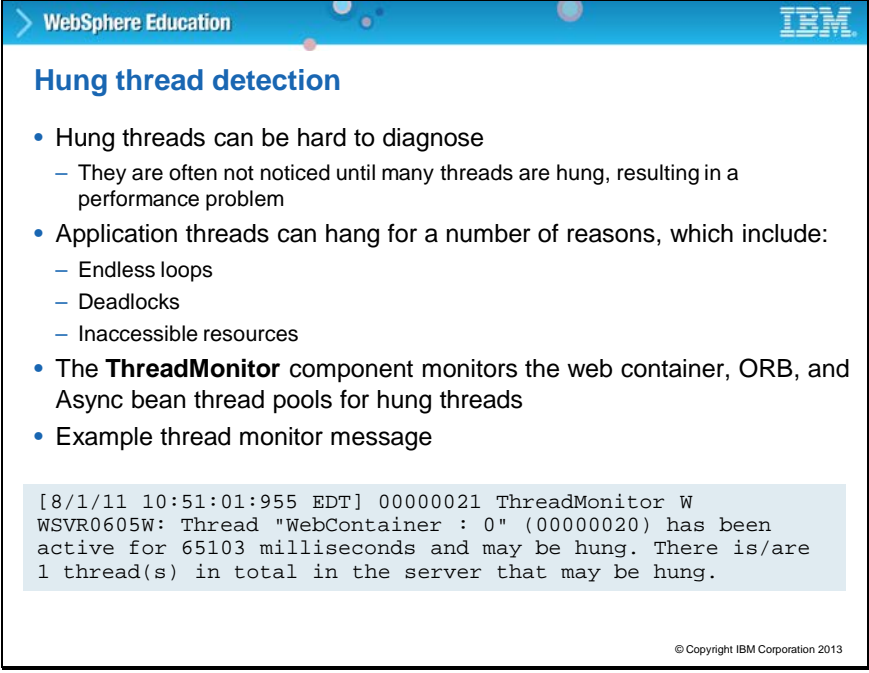
Title: Generating a JVM heap dump

The default location for a JVM heap dump is the profile root directory. You can change this location by using a generic JVM argument (-Xdump). A JVM heap dump is created when an out of memory exception is thrown. This action can be disabled by using the generic JVM argument that is shown here.

The heap dump uses portable heap dump format (phd) – a more compact format that newer tools use. You can also format the heap dump as text by using a generic JVM argument. You can trigger a heap dump by using wsadmin commands that are shown here.

IBM Support Assistant includes a tool for inspecting Java memory dumps, called the Memory Analyzer. You can start it from the IBM Support Assistant workbench. This tool can analyze several Java heap dump formats, including the standard dump formats from both the IBM and Sun Java runtime environments, and also z/OS SVC dumps. It lists and analyzes the data structures in the heap and their relationships, helping you identify the structures that are most likely responsible for memory leakage.
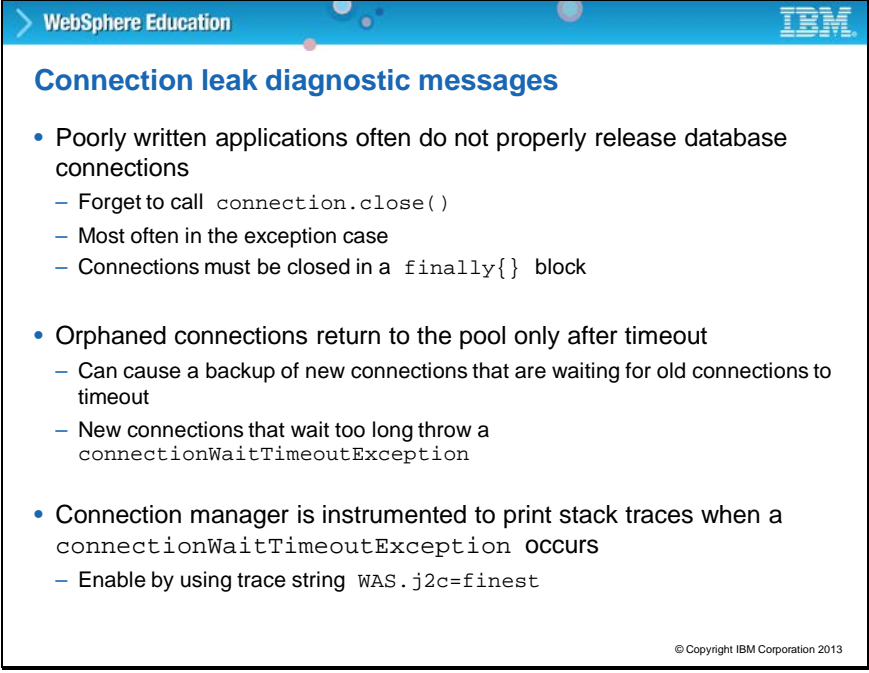
Slide 36

Title: Hung thread detection

Application threads can hang for a number of reasons, including resource contention, infinite loops, and deadlocks.

A component that is known as the ThreadMonitor monitors the web container, ORB, and Async bean thread pools for hung threads.

The thread monitor does not try to deal with the hung threads; it just issues notifications so that the administrator or developer can deal with the issues. When a hung thread is detected, three notifications are sent: a JMX notification is sent for JMX listeners, PMI Thread Pool data is updated for tools like the Tivoli Performance Viewer, and a message is written to the SystemOut log.

When the thread pool gives work to a thread, it notifies the thread monitor. The thread monitor notes the thread ID and time stamp. The thread monitor compares active threads to time stamps. Threads that are active longer than the time limit are marked "potentially hung" and notifications are sent. The performance affect on the server is minimal (less than 1%).
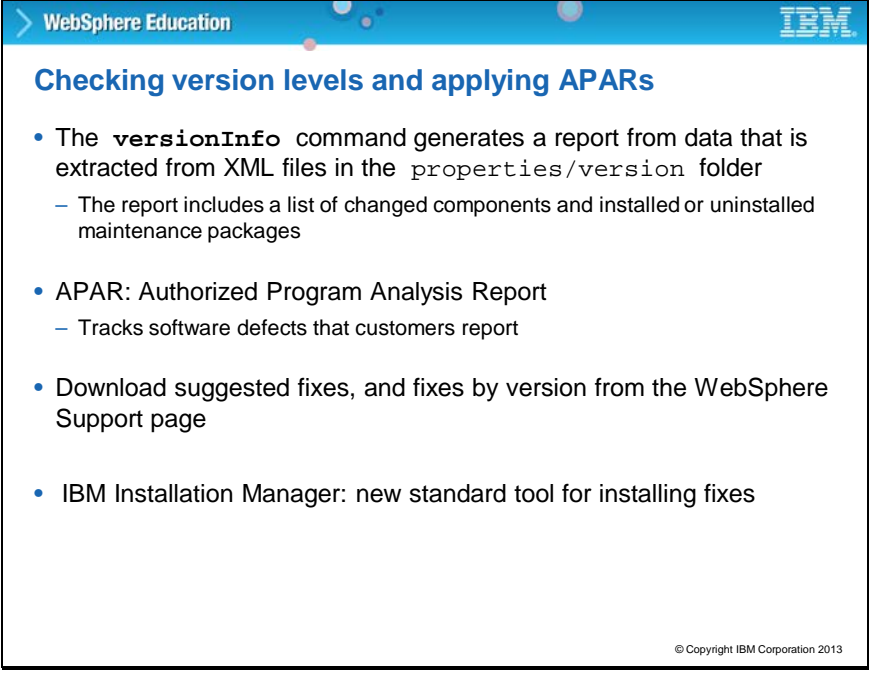
Slide 37

Title: Connection leak diagnostic messages

Applications can suffer from performance problems and even "hang" if they do not close their connections properly. Not properly using the connection.close() method often causes a connection leak. To ensure that connections are closed properly, they must be closed in a finally{} block.

WebSphere eventually times out orphaned connections and returns them to the pool, but for an application that makes frequent use of database connections, this timeout might not be enough. New connections can get queued up waiting for the database while old connections are waiting to be timed out. This behavior can halt the application, and you might see several connectionWaitExceptions in the server logs.

The leak detection that the connection manager provides has lower performance impact than tracing the connection manager. When leak detection is activated, it enables a connection manager wrapper that holds the stack trace of all getConnection() calls in a throwable object. When an exception is thrown due to waiting on a full connection pool, stack traces of *all* open connections are printed. This behavior enables you to significantly narrow your search area when you look at the application source code in an attempt to find the responsible code.

Connection leaks are difficult to diagnose because the error messages do not usually provide enough specific information about the source of the problem. Usually a source code review is needed to find points in the code where connections are not properly closed. The connection manager makes this task much easier.

37

Slide 38


Title: 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.

An APAR is an Authorized Program Analysis Report; it tracks software defects that customers report.

It is a good idea to download suggested fix packs for your version of the product from the WebSphere Support website.
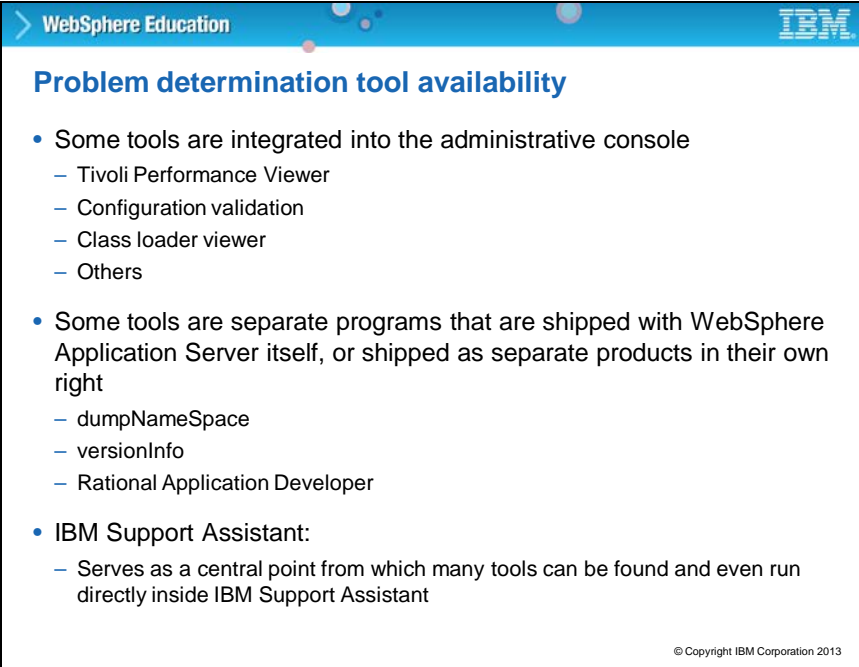
The IBM Installation Manager is the new standard tool for installing fixes.

Slide 39
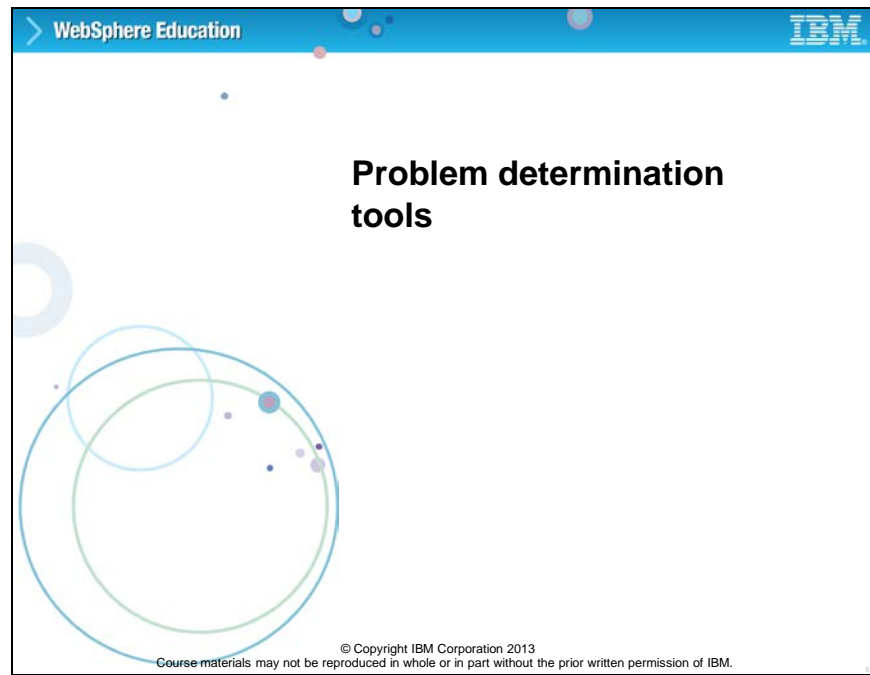
Topic: Problem determination tools

This topic describes some of the tools that can be used for problem determination.
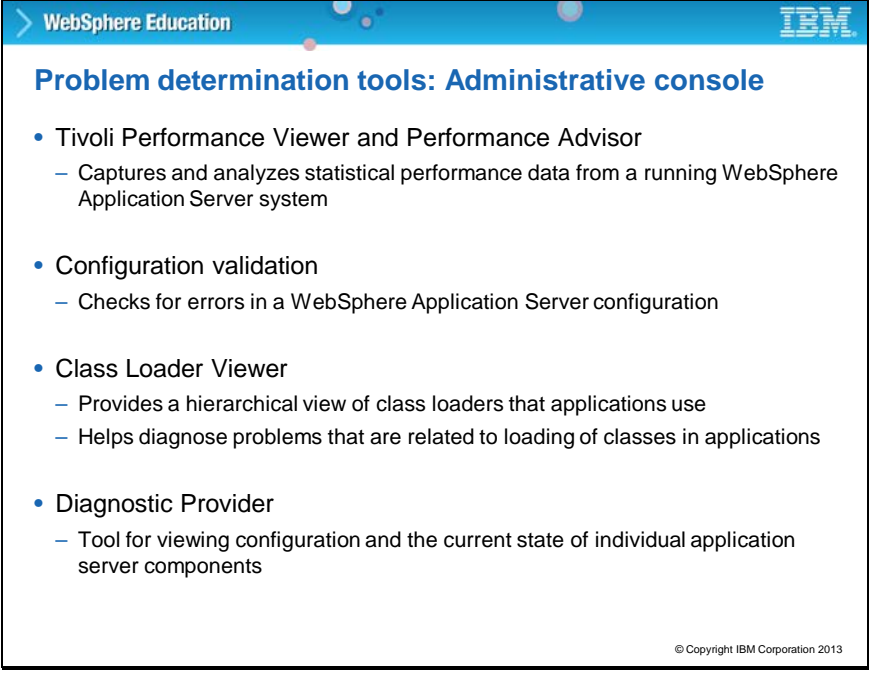
Slide 40

Title: Problem determination tool availability

In the overall process of problem determination, diagnostic data must be collected or generated, and the data must be analyzed. Various tools are available to help you collect and analyze diagnostic data for solving problems that are related to the JVM, server and application configuration, performance, and namespace issues.

Slide 41

Title: Problem determination tools: Administrative console

Several useful problem determination tools are available in the 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.

- The **Class Loader Viewer** examines a hierarchical view of class loaders that applications use and helps diagnose problems that are related to loading of classes in applications.

- The **Diagnostic Provider** is a tool for viewing configuration and the current state of individual application server components.

Slide 42

Title: Dumping the JNDI namespace

The dumpNameSpace utility shows the content of the JNDI name space. You can use it to verify the names of objects that are registered in the name space. The syntax for the command is shown here.

Slide 43
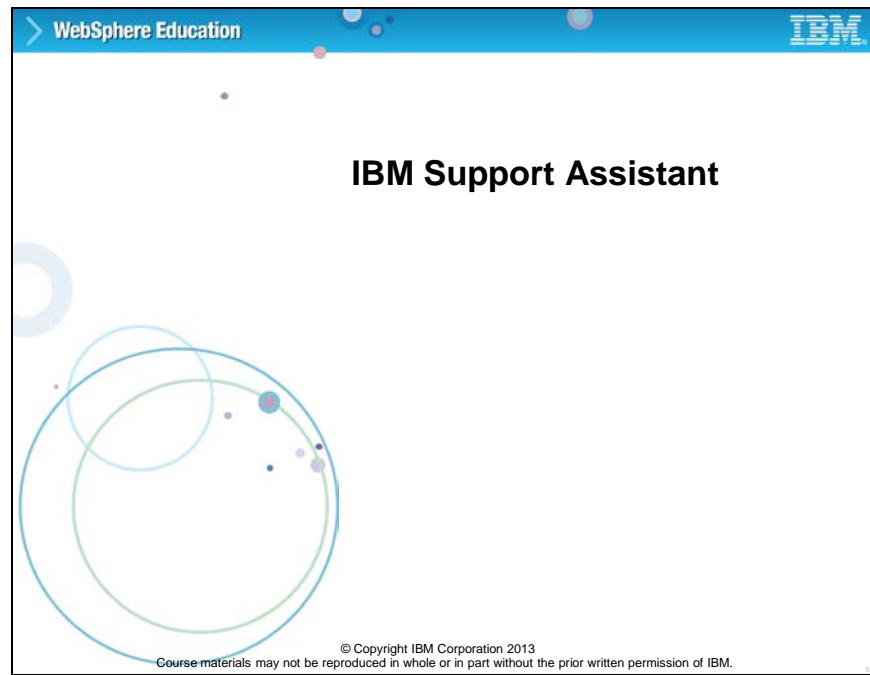
Topic: IBM Support Assistant

This topic describes how to use the IBM Support Assistant.

Slide 44

Title: What is the IBM Support Assistant?

The IBM Support Assistant is a free, stand-alone application that can be installed on any workstation. It can also be enhanced by installing plug-in modules for the IBM products you use. IBM Support Assistant provides the following benefits.

- It improves your ability to locate IBM support, development, and educational information through a federated search interface (one search: multiple resources).

- It provides quick access to the IBM Education Assistant and key product education roadmaps.

- It simplifies access to IBM product home pages, product support pages, and product forums or newsgroups, through convenient links.

- It saves time in submitting problems to IBM Support by collecting key information, then electronically creating a problem management record (PMR) from within IBM Support Assistant. If a problem management record (PMR) must be opened, IBM Support Assistant helps with:

- Gathering support information that is based on problem type

- Creating and updating the problem report

- Tracking your electronic problem report

- It includes a support tool framework, allowing for the easy installation of support tools that are associated with different IBM products.

44

- It provides a framework for IBM software products to deliver customized self-help information into the different tools within it. You can customize your IBM Support Assistant client by using the built-in update capability to find and install new product features or support tools.

Slide 45
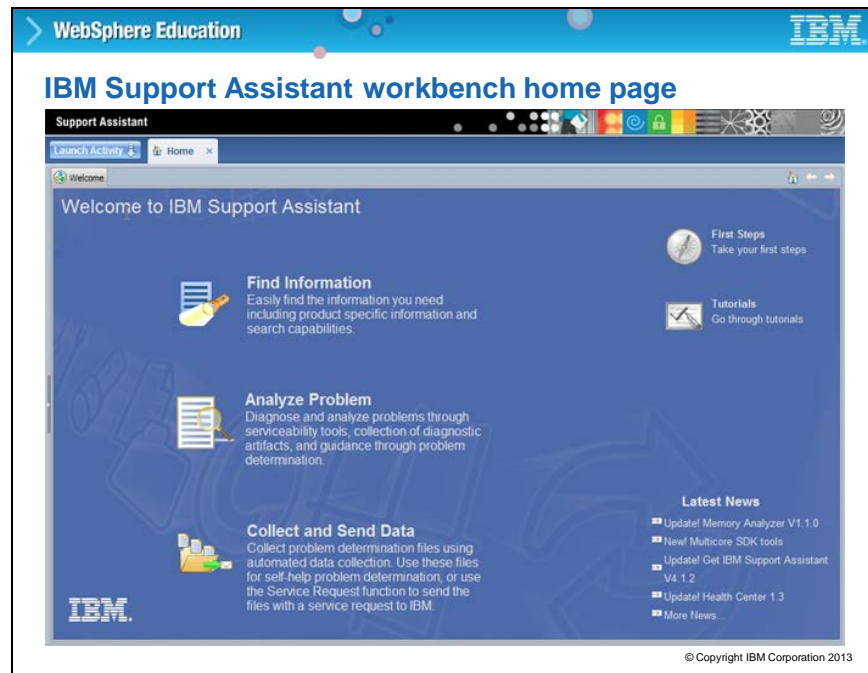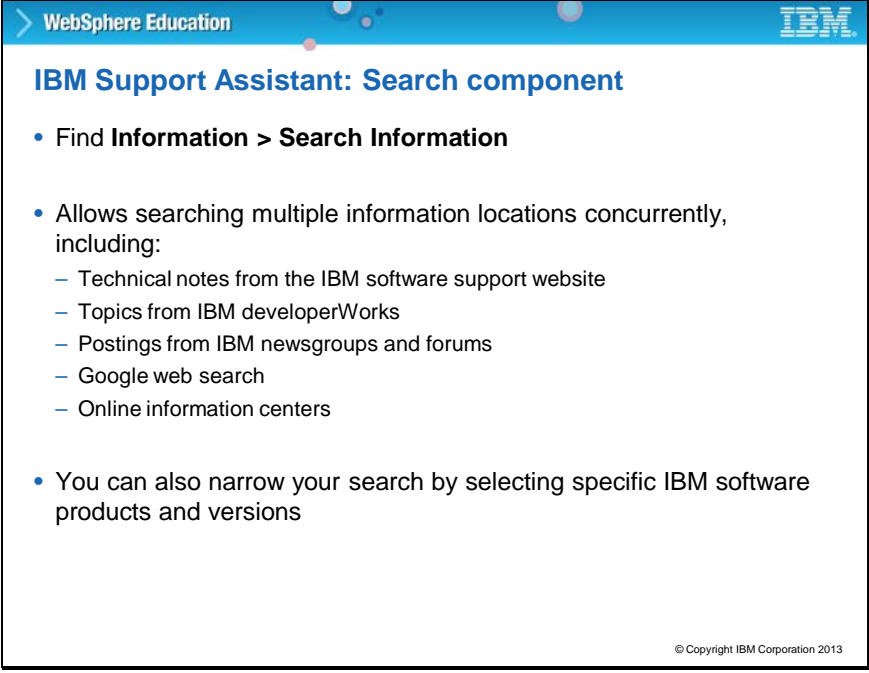
Title: IBM Support Assistant workbench home page

To download and install ISA, go to the IBM software support portal and choose the IBM Support Assistant download link for your platform. Log in by using your IBM ID. If you do not already have an IBM ID, you can register for one. After downloading the compressed archive files, extract the archive to a temporary directory and then run the installer program.

This screen shows the ISA workbench home page.

Slide 46

Title: IBM Support Assistant: Search component

In ISA, search options are provided to help you narrow the scope of your search. The search location that you choose determines what search options are available. For example, if you select Google web search, search options do not apply. The **IBM developerWorks** and **IBM newsgroups and forums** search locations allow you to narrow your search to specific product areas.

The **IBM software support documents** search location offers the most options. You can select specific products and versions, and you can select specific types of documents.

The online information centers are only available for products that enable this feature. Most IBM software products have an information center to search.

Slide 47

Title: IBM Support Assistant: Tools component

The tools component is a way to provide support tools to you. These tools can be used diagnostically to identify problems you are having. Some tools can also be used as a preventive measure to stop a problem before it occurs.

The component provides a way to use support-related tools to identify problems or health-checking tools to prevent problems. Tools are installed by using the Updater component of IBM Support Assistant. A tool is started in a new window by clicking its name.

Slide 48

Title: IBM Support Assistant: Collect Data component

Collecting data is optional, but proactive collection of diagnostic data can help you troubleshoot problems with IBM software and help IBM resolve your service requests faster. If you are troubleshooting a problem on your own or intend to submit a service request, collect data first. You can store the collected data in a case (a placeholder for all the items that are related to a particular issue) by using the Case Manager. Using a case helps to organize your problem determination efforts and allows you to send the collected data to IBM when you create a service request.

Slide 49

Title: IBM Support Assistant: Collect and send data

Collecting data is an important part of submitting a problem report because IBM can use it to resolve your problem report more quickly.

When you click **Collect Data**, you are asked to select the product for which you would like to collect data. You are prompted to select a problem type as well.

Depending on the product you select, you might be asked for other information. When you select a product and provide the required information, press **Collect**.

For most problem types, IBM Support Assistant collects data for the product you select and stores it in a JAR file. For some problem types, the information that is collected results in a report file or a simple message to the user. You can collect data for more than one product at a time if wanted.

To manage problem reports, you must log in. After you are logged in, you can:

- Open a new problem report
- View existing problem reports
- Attach a file to a new or existing problem report

To log in, you must provide your IBM ID, password, IBM customer number, and country. The login process also verifies that you have an IBM service contract and that your ID is on the Authorized Callers list.

After you log in, the service component shows you any problem reports that you submitted. You can see these problem reports at any time by clicking **List Problem Reports**.

50

Slide 50
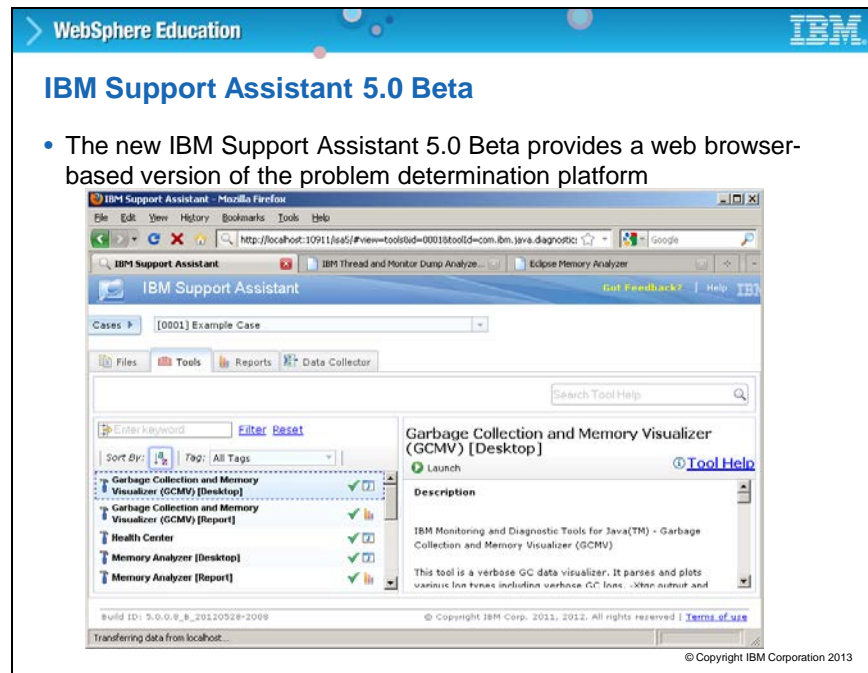
Title: IBM Support Assistant 5.0 Beta
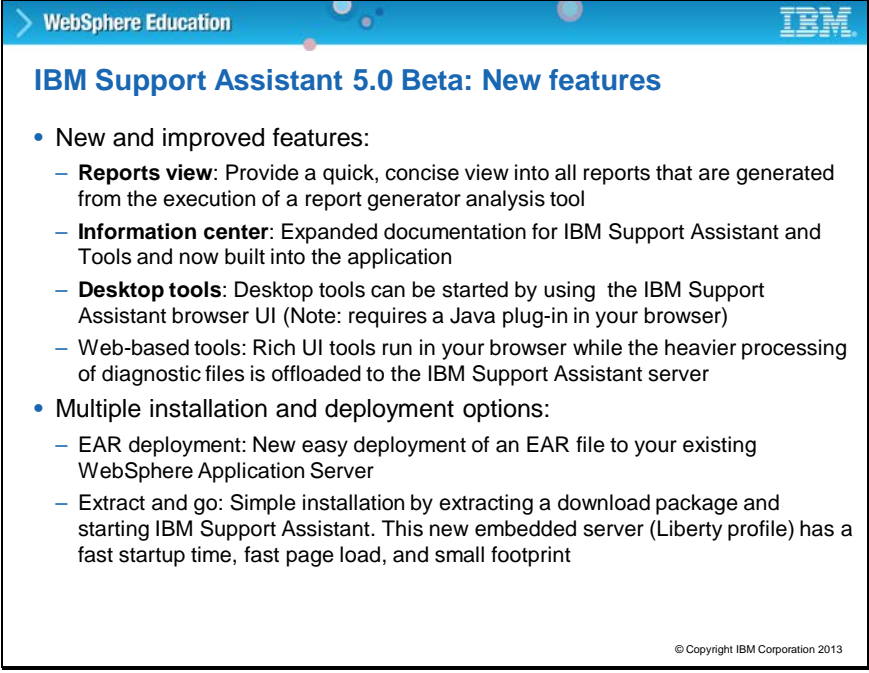
You can download IBM Support Assistant 5.0 Beta from ISA website.

In this Beta release, new problem determination tools are included and the number of tools will continue to grow in future releases. Report-generating tools from the technology preview, along with the new addition of Desktop tools (by using Java Webstart) and web-based tools, include the following tools:

- Garbage Collection and Memory Visualizer (GCMV)
- Health Center
- Memory Analyzer
- Thread and Monitor Dump Analyzer (TMDA)
- WebSphere Application Server Configuration Visualizer
- And several others

To learn more about the tools in this release, select the Help link in the IBM Support Assistant UI to access the new information center documentation.

51

Slide 51

WebSphere Education IBM

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
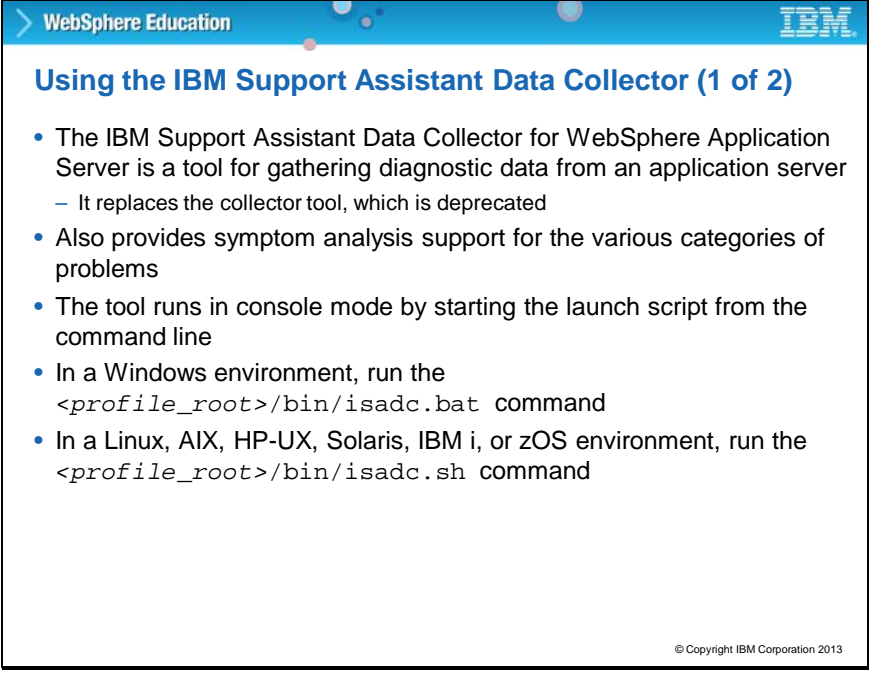
© Copyright IBM Corporation 2013

Title: IBM Support Assistant 5.0 Beta: New features

IBM Support Assistant is provided with two deployment options to meet your needs:

If you have an existing WebSphere Application Server, you can install IBM Support Assistant as an enterprise application (EAR) module. Tools are provided as Java Platform, Enterprise Edition web modules.

IBM Support Assistant 5.0 Beta is also provided with an embedded server if you are looking for an all-in-one package that allows you to extract and go.

52

Slide 52

Title: Using IBM Support Assistant Data Collector (1 of 2)

The IBM Support Assistant Data Collector for WebSphere Application Server tool focuses on automatic collection of problem data. It also provides symptom analysis support for the various categories of problems that by IBM software products encounter. Information pertinent to a type of problem is collected to help identify the origin of the problem under investigation. The tool assists customers by reducing the amount of time it takes to reproduce a problem with the proper RAS tracing levels set. Also, the tool helps customers by reducing the effort that is required to send the appropriate log information to IBM Support.

Slide 53
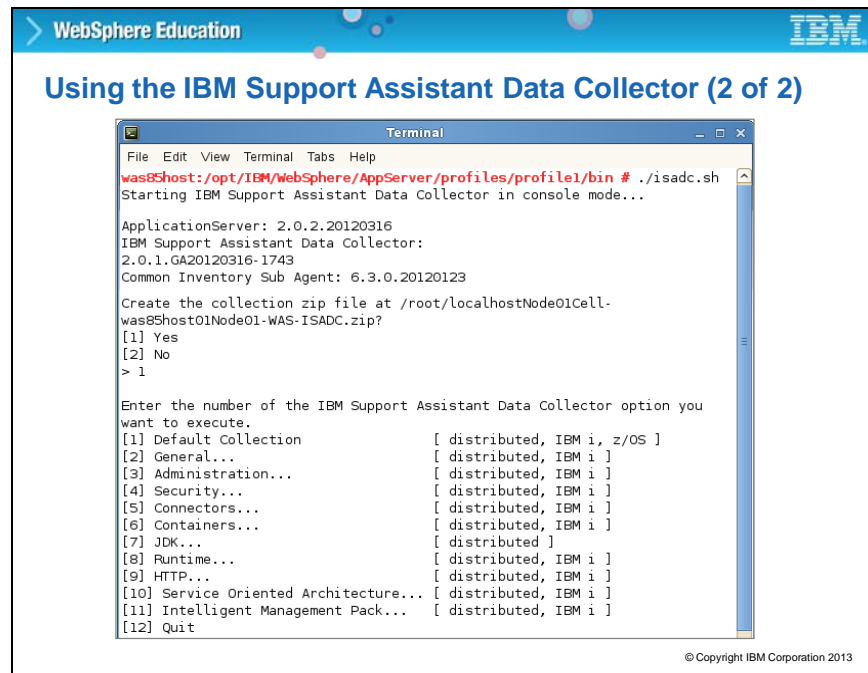
Title: Using IBM Support Assistant Data Collector (2 of 2)

Run the IBM Support Assistant Data Collector tool with the user ID for which you configured your WebSphere Server instance. Depending on what collector you are running, you might be asked for more information to complete the data collection activities. A script might require more configuration information, information about the sequence of events that leads up to the problem you are dealing with, or for your preferences about how it completes the collection. At each step, the choices are presented as numbered lists and you input the number of your selection and press the enter key. When input is required, prompts are displayed at which you enter your response and press the enter key. You can find collection details for each WebSphere Application Server problem type in their corresponding MustGather documents.

Slide 54

Title: 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