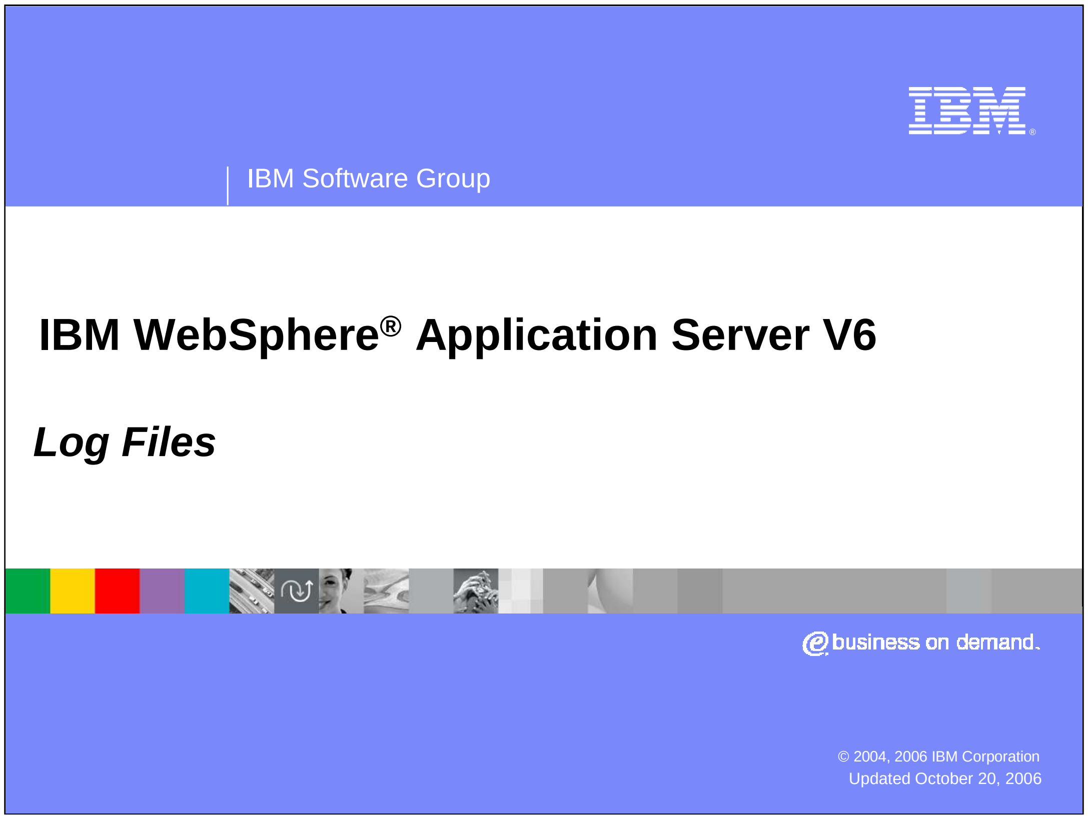


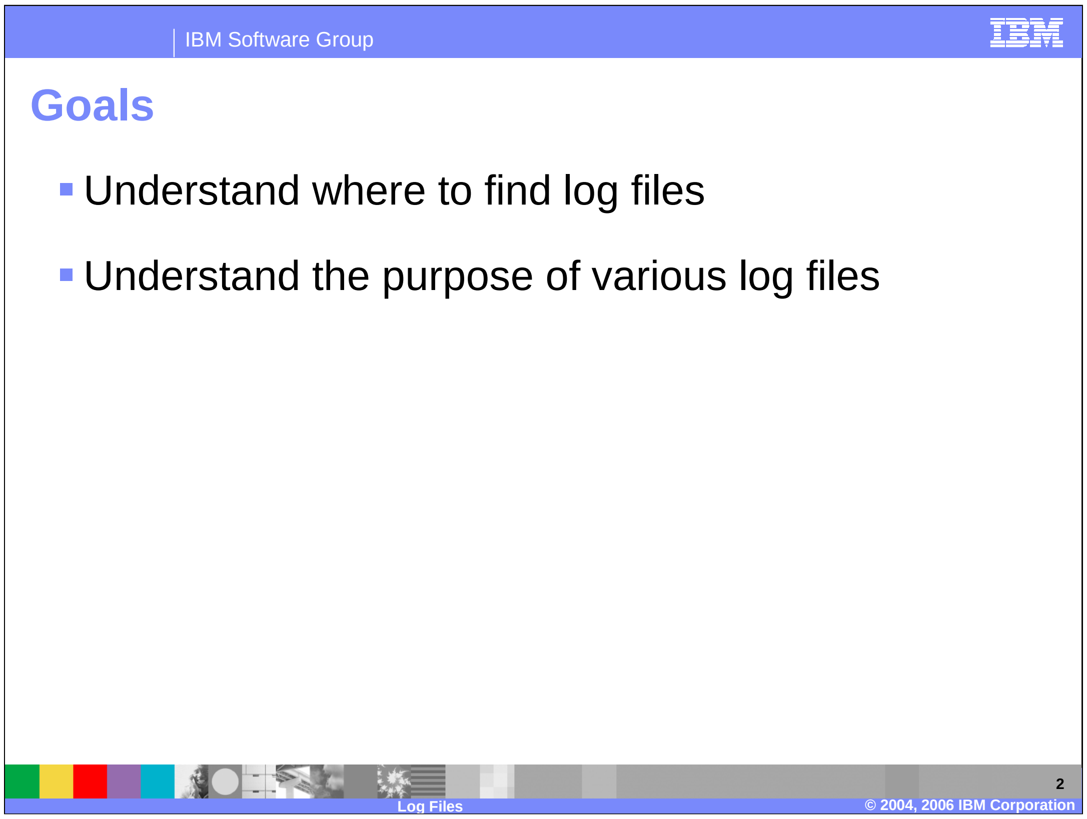
IBM Software Group

IBM WebSphere® Application Server V6

Log Files

@business on demand.

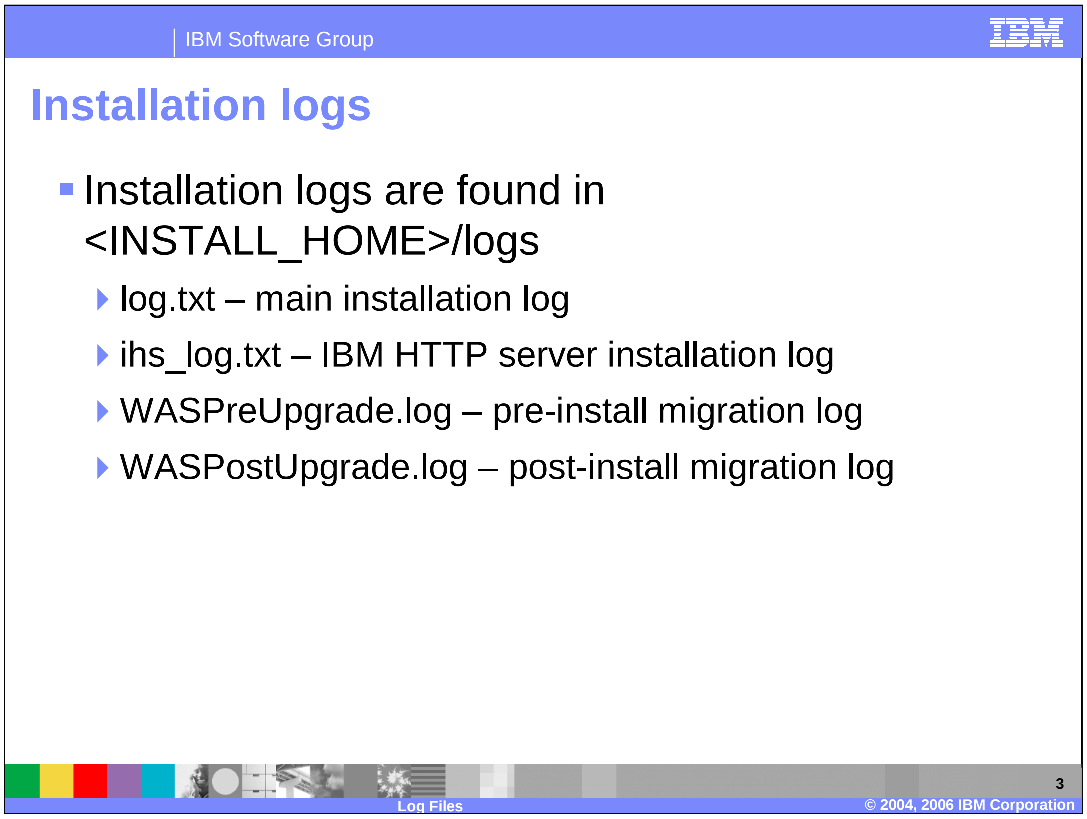
This presentation will focus on the basic log files and their purposes.



Goals

- Understand where to find log files
- Understand the purpose of various log files

The goals for this presentation are to help you understand where to find the different log files created by WebSphere Application Server and to teach you about the kind of information you look for in them.


IBM Software Group

Installation logs

- Installation logs are found in <INSTALL_HOME>/logs
  - log.txt – main installation log
  - ihs_log.txt – IBM HTTP server installation log
  - WASPreUpgrade.log – pre-install migration log
  - WASPostUpgrade.log – post-install migration log

3

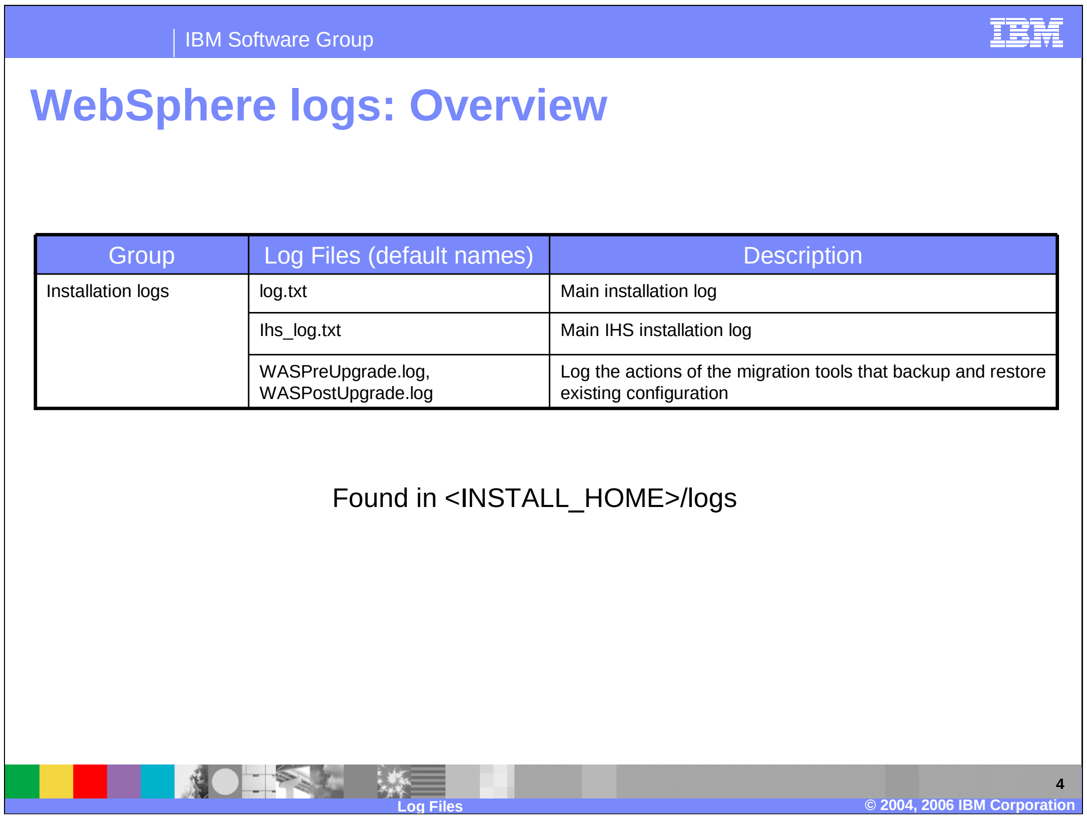
Log Files

© 2004, 2006 IBM Corporation

The main installation log is log.txt. This file shows the progress of the installation, and should be consulted first if your installation fails. You will be able to see what stage of the installation is failing, and hopefully deduce the cause. If your installation fails before the root installation directory can even be created, log.txt will be located in your system's temp directory, such as "/tmp".

You should look at ihs_log.txt if the installation of the IBM HTTP Server fails.

WasPreUpgrade.log and WASPostUpgrade.log are only created if you are using the migration tools provided with WebSphere Application Server to migrate your installation from a previous release.



WebSphere logs: Overview

Group	Log Files (default names)	Description
Installation logs	log.txt	Main installation log
	Ihs_log.txt	Main IHS installation log
	WASPreUpgrade.log, WASPostUpgrade.log	Log the actions of the migration tools that backup and restore existing configuration

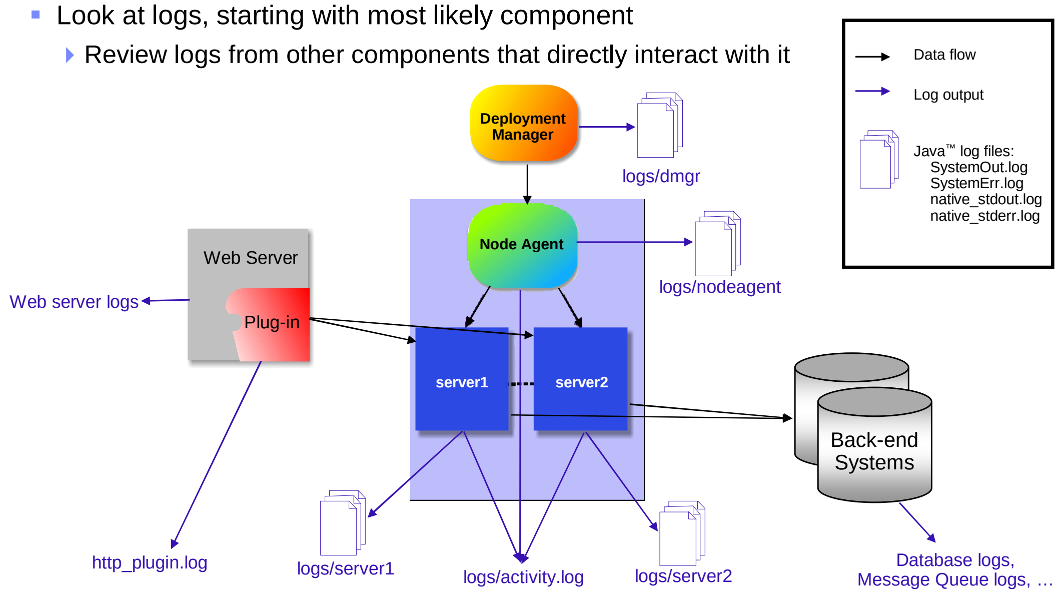
Found in <INSTALL_HOME>/logs

This table is provided for your reference, to help you locate the installation logs

IBM Software Group

Components and log files

- Look at logs, starting with most likely component
  - Review logs from other components that directly interact with it



5

Log Files

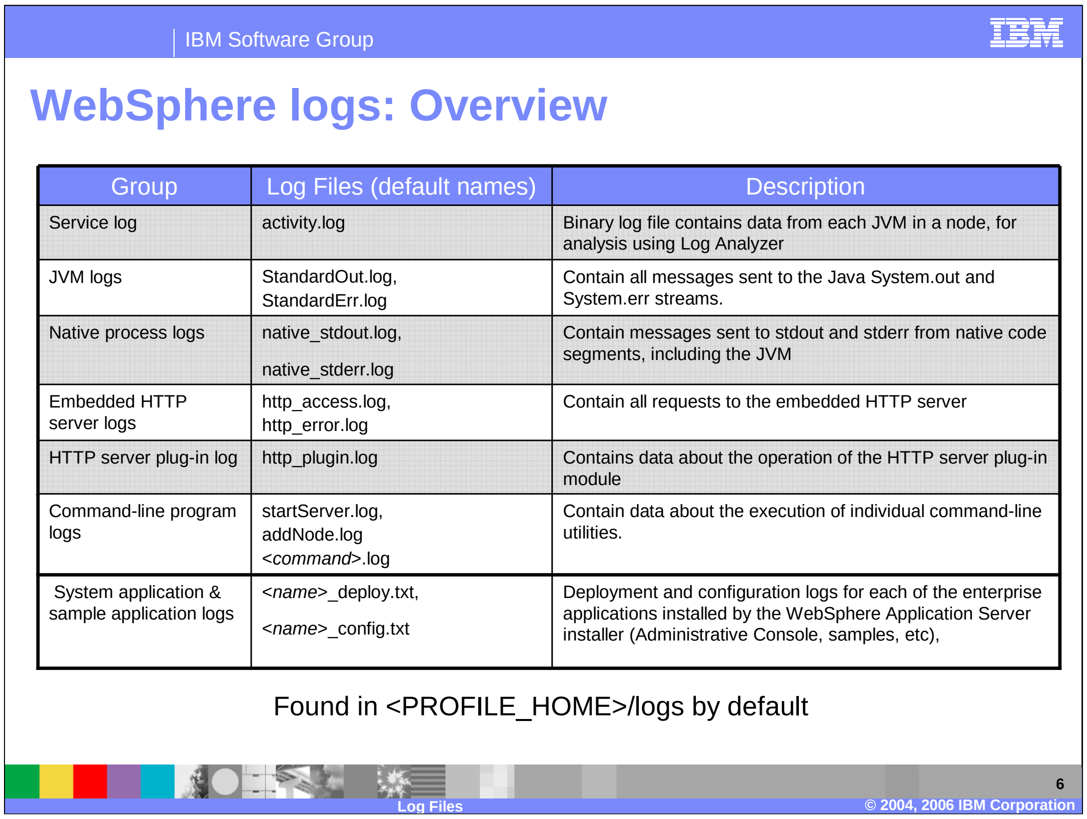
© 2004, 2006 IBM Corporation

During runtime, each WebSphere Application Server process creates log files that report the actions being performed. Each of these logs can be very important to troubleshooting a problem.

While there can be several pieces to a complete production environment, this lecture will be specifically discussing logs created by WebSphere Application Server itself.

Even though they will not be covered here, logs from other systems can be important (for example, Web server, database, or firewall logs). Consult your product’s documentation for information on its log files.

There are also many log files that are not shown in this picture. The logs shown in this picture are the logs that you should look at first when dealing with a problem during runtime. The other logs will be discussed later.


IBM Software Group

WebSphere logs: Overview

Group	Log Files (default names)	Description
Service log	activity.log	Binary log file contains data from each JVM in a node, for analysis using Log Analyzer
JVM logs	StandardOut.log, StandardErr.log	Contain all messages sent to the Java System.out and System.err streams.
Native process logs	native_stdout.log, native_stderr.log	Contain messages sent to stdout and stderr from native code segments, including the JVM
Embedded HTTP server logs	http_access.log, http_error.log	Contain all requests to the embedded HTTP server
HTTP server plug-in log	http_plugin.log	Contains data about the operation of the HTTP server plug-in module
Command-line program logs	startServer.log, addNode.log *<command>*.log	Contain data about the execution of individual command-line utilities.
System application & sample application logs	*<name>*_deploy.txt, *<name>*_config.txt	Deployment and configuration logs for each of the enterprise applications installed by the WebSphere Application Server installer (Administrative Console, samples, etc),

Found in <PROFILE_HOME>/logs by default

6

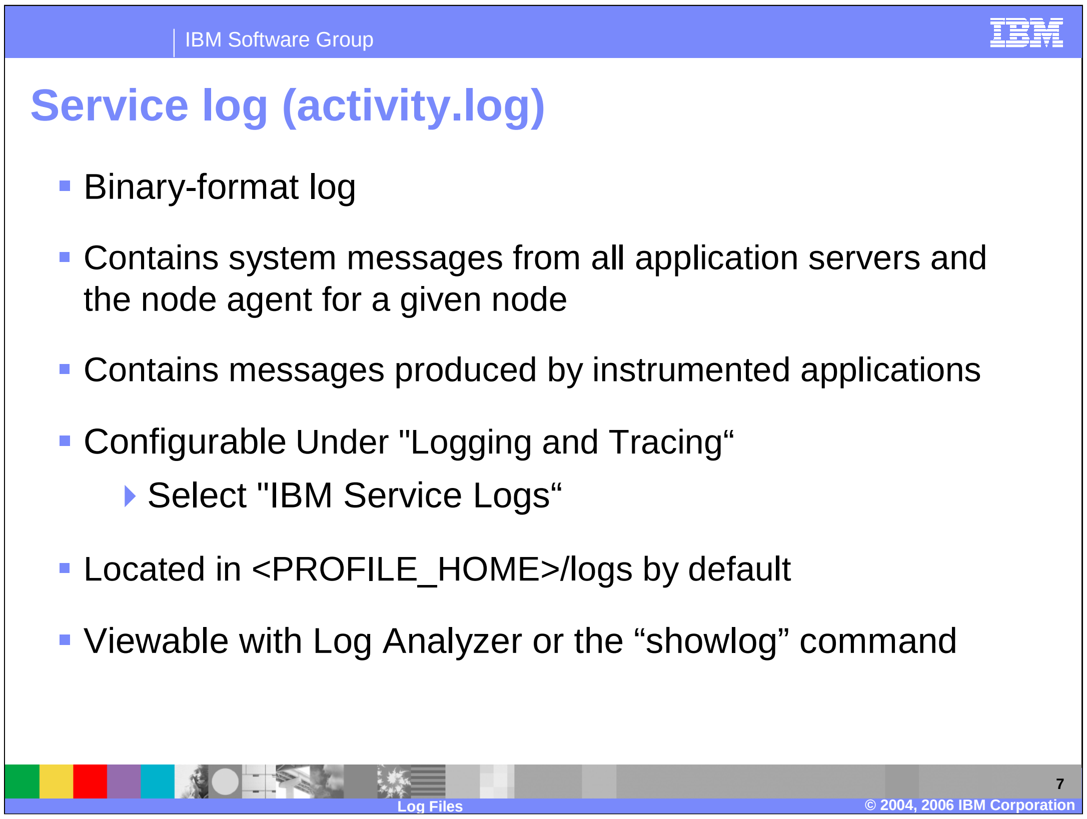
Log Files

© 2004, 2006 IBM Corporation

The logs shown on this page will exist for each profile, and as such will exist in the particular profile's logs directory. Other than activity.log, which only exists once per node, each server on your node will have its own copy of these runtime log files.

The JVM logs are the most useful of the runtime logs. They contain information about the server runtime, and any messages that the application writes to System.out or System.err. You will also find exception and stack trace information in these logs. The native process logs contain information that gets logged by native code, such as the JVM itself, or some parts of the security implementation.

The other logs shown in this table are relatively self-explanatory; they correspond to a particular component or command-line program.

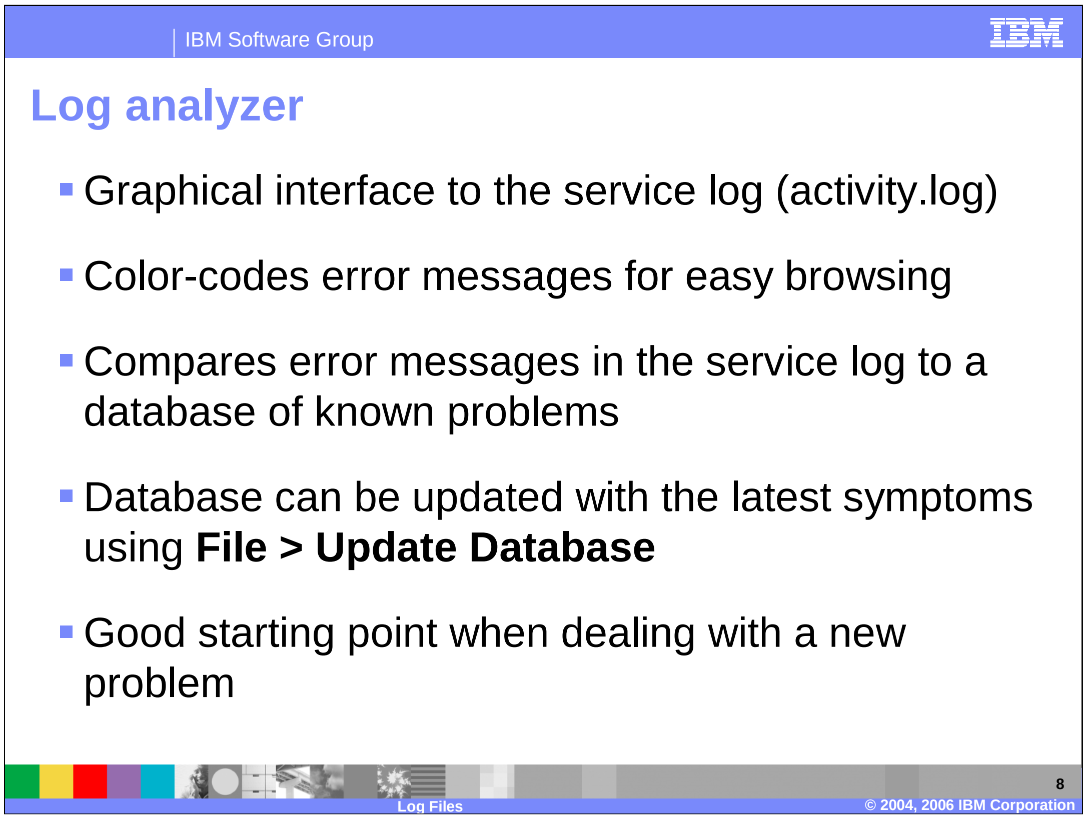


Service log (activity.log)

- Binary-format log
- Contains system messages from all application servers and the node agent for a given node
- Contains messages produced by instrumented applications
- Configurable Under "Logging and Tracing“
  - Select "IBM Service Logs“
- Located in <PROFILE_HOME>/logs by default
- Viewable with Log Analyzer or the “showlog” command

The service log is a binary-format log that contains more or less the same messages as the SystemOut JVM log, plus a few extra serviceability messages.

The benefit of the service log is that it can be used in conjunction with the Log Analyzer, which can compare logged messages with a “symptom database”. The symptom database correlates messages to known problems. Viewing your service log with Log Analyzer is a good first step in problem determination, because it can compare your situation to a list of known WebSphere problems.


IBM Software Group

Log analyzer

- Graphical interface to the service log (activity.log)
- Color-codes error messages for easy browsing
- Compares error messages in the service log to a database of known problems
- Database can be updated with the latest symptoms using **File > Update Database**
- Good starting point when dealing with a new problem

8

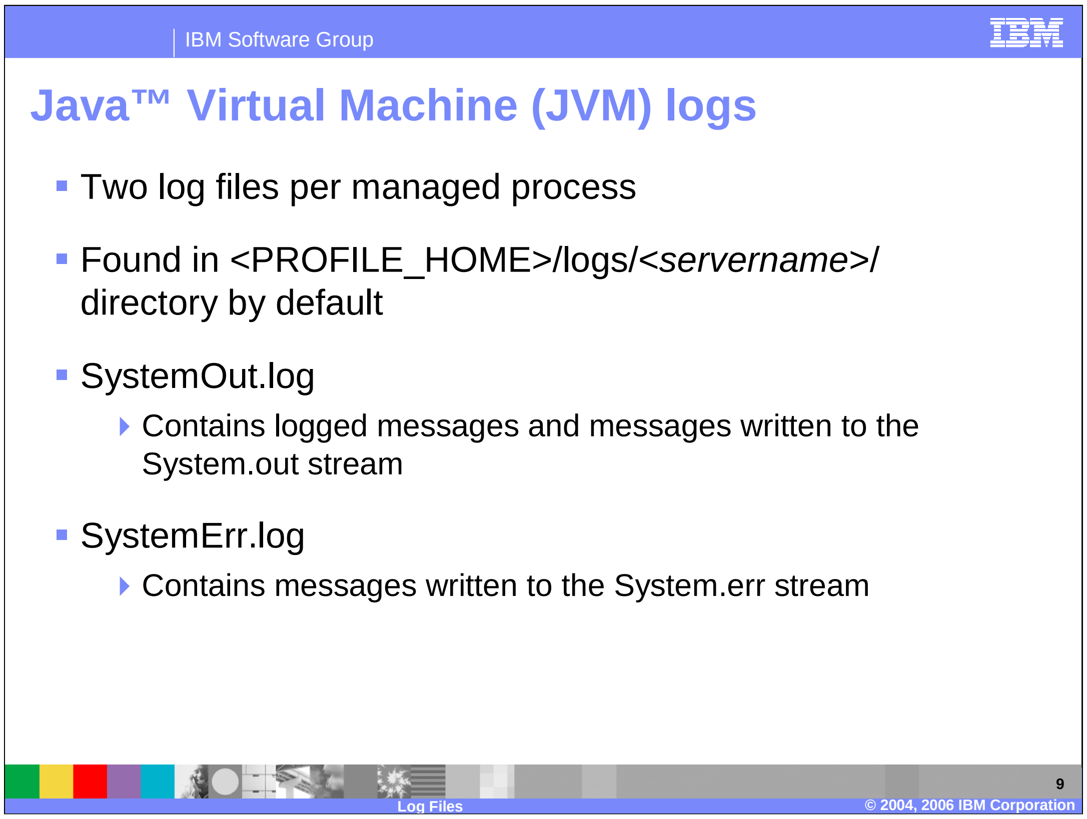
Log Files

© 2004, 2006 IBM Corporation

Log Analyzer is a graphical interface to the service log.

You can use it to browse the log and quickly view messages by severity, and get detailed information about individual messages.

Most importantly, Log Analyzer contains a database of known problems against which you can compare your error messages. When you are facing a new problem, this can be a good place to start, because the database may contain valuable information about your problem. You can update the database over the Internet to add the latest data using the “Update Database” item in the File menu.

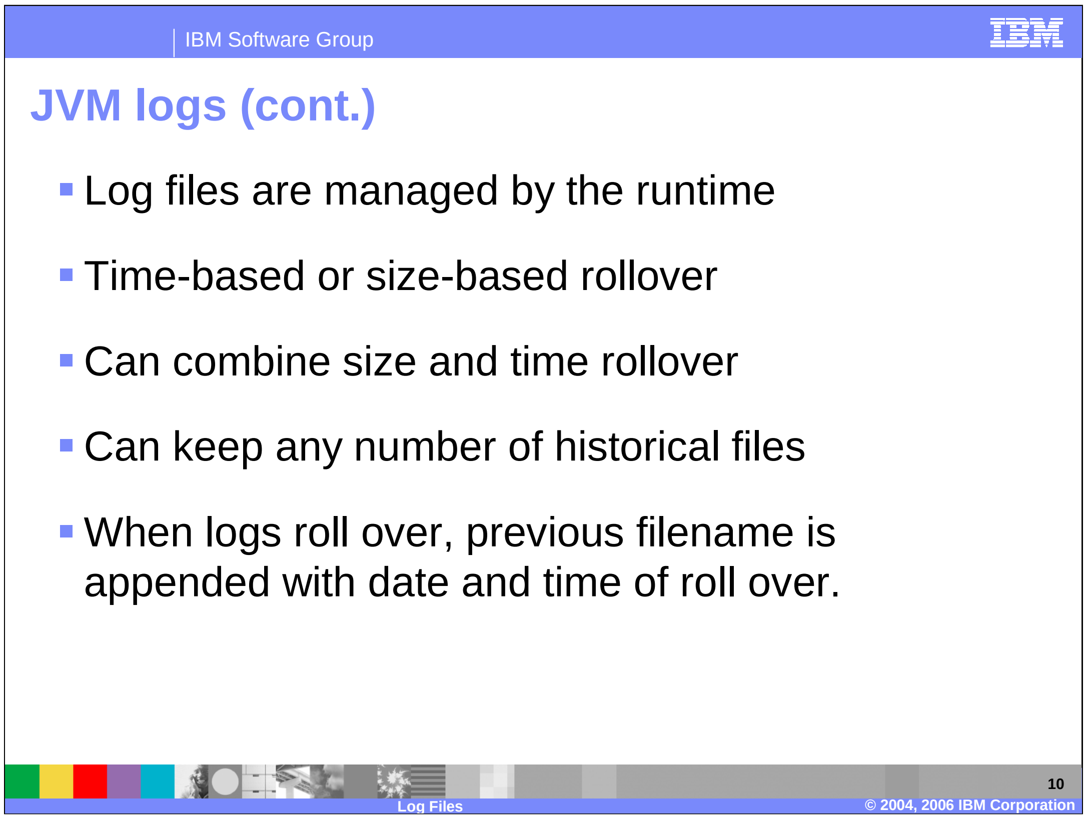


Java™ Virtual Machine (JVM) logs

- Two log files per managed process
- Found in <PROFILE_HOME>/logs/<*servername*>/ directory by default
- SystemOut.log
  - Contains logged messages and messages written to the System.out stream
- SystemErr.log
  - Contains messages written to the System.err stream

The most commonly used logs are the JVM logs, often referred to as “standard out” and “standard error”. They contain messages written to the System.out and System.err streams, respectively. Each Java process has its own JVM logs. Exceptions are written to these logs from Exception.printStackTrace().

The JVM logs are a good place to look for detailed information when there is a problem with an Application Server. Node Agent and Deployment Manager processes also write to JVM logs. You should look at these if the Node Agent or Deployment Manager is unable to start, similar to the trace file in older versions of WebSphere Application Server.

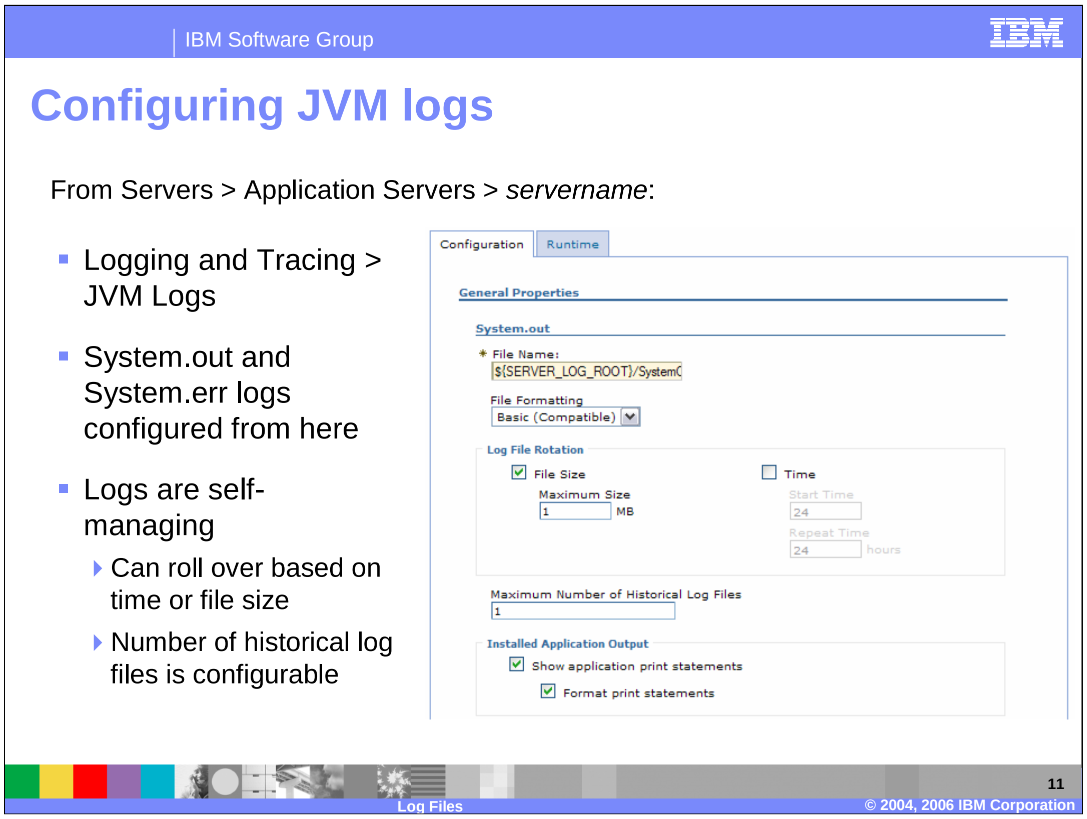


JVM logs (cont.)

- Log files are managed by the runtime
- Time-based or size-based rollover
- Can combine size and time rollover
- Can keep any number of historical files
- When logs roll over, previous filename is appended with date and time of roll over.

A log can be configured to roll over at specific time intervals, or when the file reaches a certain size. The two can also be combined. You could, for example, configure the log to roll over each day at midnight, unless the file grows larger than 10MB, in which case it rolls over immediately.

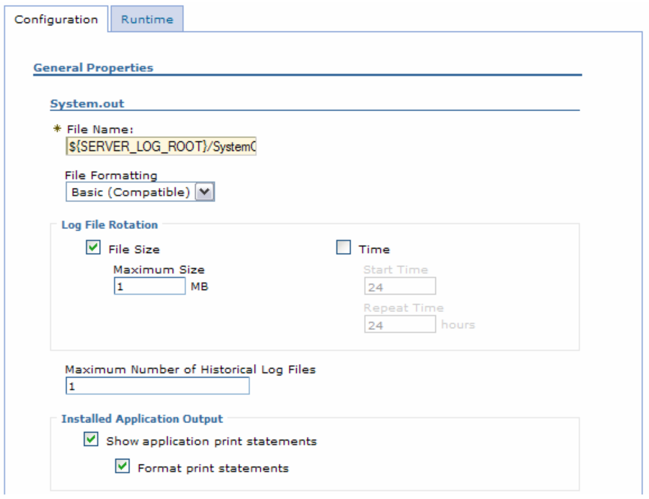
You can also specify how many "historical", meaning previously rolled-over, log files to keep.



Configuring JVM logs

From Servers > Application Servers > *servername*:

- Logging and Tracing > JVM Logs
- System.out and System.err logs configured from here
- Logs are self-managing
  - Can roll over based on time or file size
  - Number of historical log files is configurable



General Properties

System.out

* File Name: ${SERVER_LOG_ROOT}/SystemO

File Formatting: Basic (Compatible)

Log File Rotation

✓ File Size

Maximum Size: 1 MB

☐ Time

Start Time: 24

Repeat Time: 24 hours

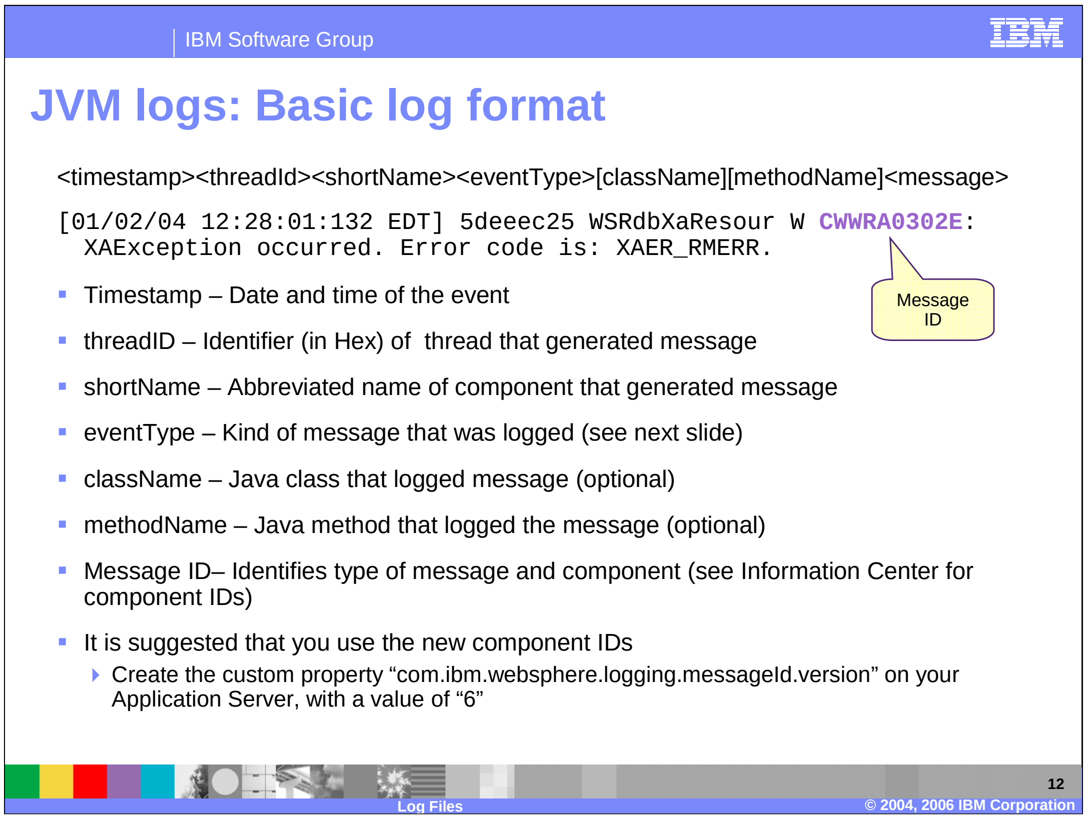
Maximum Number of Historical Log Files: 1

Installed Application Output

✓ Show application print statements

✓ Format print statements

From the server’s main configuration panel, click on “Logging and Tracing”, then “JVM Logs” to reach the JVM Log configuration panel. This screen enables you to set up the rollover features that were described on the previous slide. The “runtime” tab on this panel allows you to view the JVM logs remotely through your web browser.



JVM logs: Basic log format

<timestamp><threadId><shortName><eventType>[className][methodName]<message>

```
[01/02/04 12:28:01:132 EDT] 5deeec25 WSRdbXaResour W CWWRA0302E:
  XAException occurred. Error code is: XAER_RMERR.
```

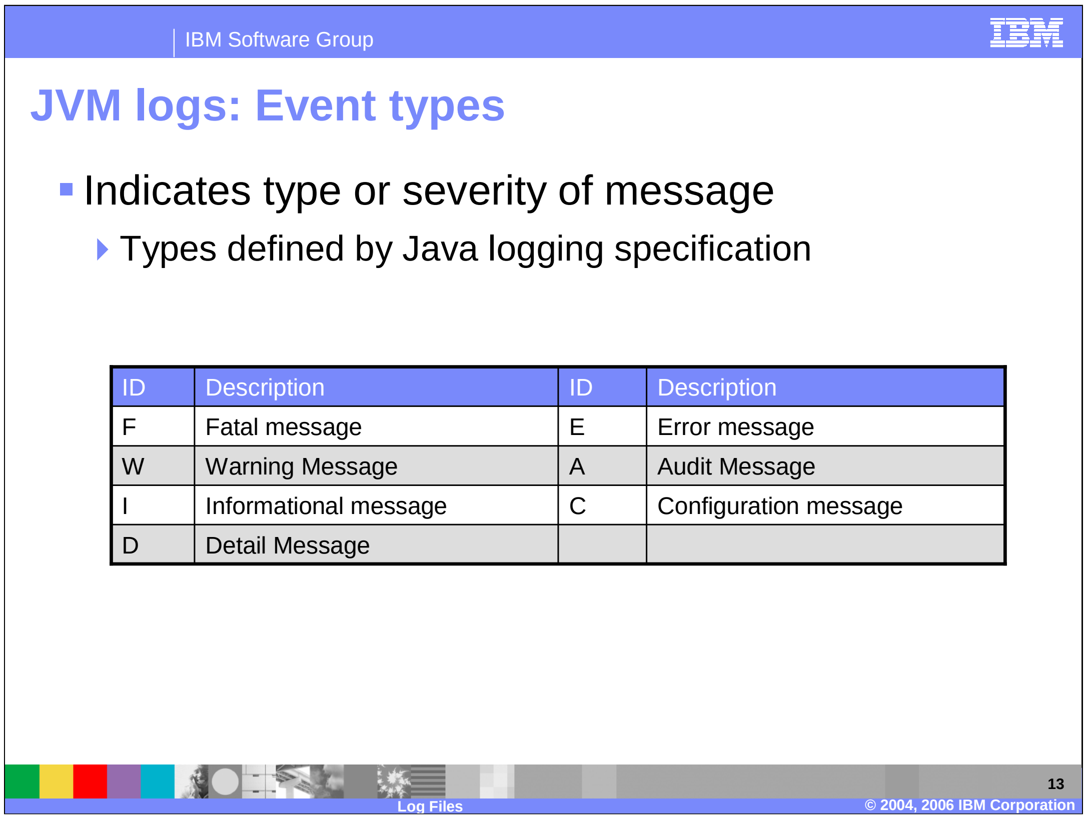
- Timestamp – Date and time of the event
- threadID – Identifier (in Hex) of thread that generated message
- shortName – Abbreviated name of component that generated message
- eventType – Kind of message that was logged (see next slide)
- className – Java class that logged message (optional)
- methodName – Java method that logged the message (optional)
- Message ID– Identifies type of message and component (see Information Center for component IDs)
- It is suggested that you use the new component IDs
  - Create the custom property “com.ibm.websphere.logging.messageId.version” on your Application Server, with a value of “6”

Messages in the JVM logs use the format shown here. The log messages are broken down into the following elements:

The timestamp is the date and time of the event. The thread ID is an identifier that represents the thread that generated the message. ShortName represents the abbreviated name of the component that generated the message. EventType tells you what kind of message was logged. The different event types are shown in the table on the next slide. ClassName is the name of the Java class that logged the message. MethodName lists the Java method that logged the message. Message ID identifies the type of message and the component. See the Information Center for a complete list of component IDs.

New component identifiers have been created for V6, in an effort to adopt a consistent format across IBM software products. But all are not enabled by default. It is recommended that you use the new component IDs, which can be enabled by setting a custom property on your Application Server. The example shown here uses the new format.

In the new format, all WebSphere messages begin with “CW”, which stands for “Components for WebSphere”. In the example highlighted above, the “WRA” indicated that this message was generated by the Resource Adapter. A mapping of old-to-new component IDs is provided in the Information Center. The numeric portion of the message identifies the particular error that occurred, and the trailing “E” indicates that an error occurred. This information will of course be followed by the actual text of the message, as shown in the example. If you are using the new component IDs, any custom scripts that you may have written to parse log files using old IDs will need to be updated.



JVM logs: Event types

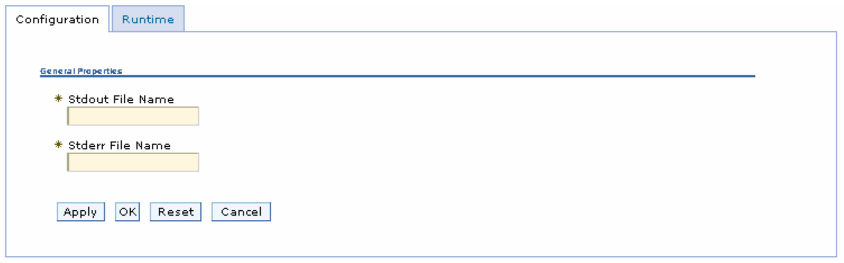
- Indicates type or severity of message
  - Types defined by Java logging specification

ID	Description	ID	Description
F	Fatal message	E	Error message
W	Warning Message	A	Audit Message
I	Informational message	C	Configuration message
D	Detail Message		

The event type indicates the type or severity of the message being logged. Some other event types are used by WebSphere diagnostic tracing, and are listed in another table later in this presentation.

Native process logs

- Native logs contain stderr and stdout messages from native code
- Log files do not rollover
  - Relatively few messages written to these logs
- Filenames are configurable under the "Process Logs" link in the Application Server's "Logging and Tracing" panel

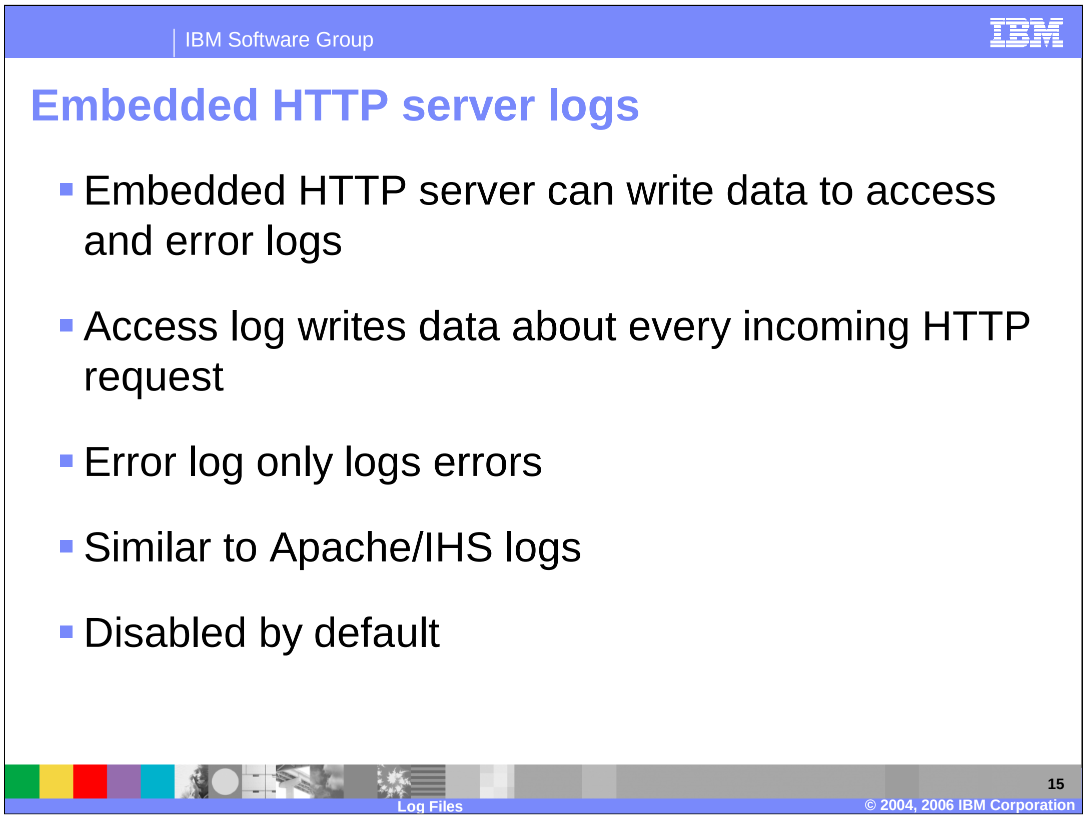


* Stdout File Name

* Stderr File Name

The native process logs contain messages written by native code segments, including the JVM itself. Given the relatively small amount of native code compared to Java code, very few messages are written to these logs. Since the files are small, they do not have rollover capability.

One component that does use a reasonable amount of native code is the security subsystem. If you are having security-related problems, you should take a look at the native logs.


IBM Software Group

Embedded HTTP server logs

- Embedded HTTP server can write data to access and error logs
- Access log writes data about every incoming HTTP request
- Error log only logs errors
- Similar to Apache/IHS logs
- Disabled by default

15

Log Files

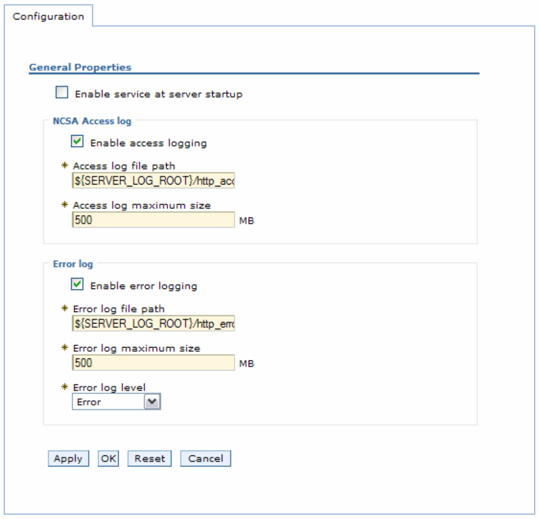
© 2004, 2006 IBM Corporation

The embedded HTTP server can write to access and error logs in the NCSA common or combined formats that are used by Apache and IHS.

These logs are disabled by default, but can be enabled using the Administrative Console or wsadmin. They can be particularly useful when you want to verify that a request is reaching the Application Server, or track the progress of a particular request as it moves through your environment.

Embedded HTTP server logs (cont.)

- New Admin. Console panels for configuring embedded HTTP Server logs (access & error)
  - Previously could only be configured by setting custom properties
- 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



☐ Enable service at server startup

NCSA Access log

☑ Enable access logging

* Access log file path: ${SERVER_LOG_ROOT}/http_acc

* Access log maximum size: 500 MB

Error log

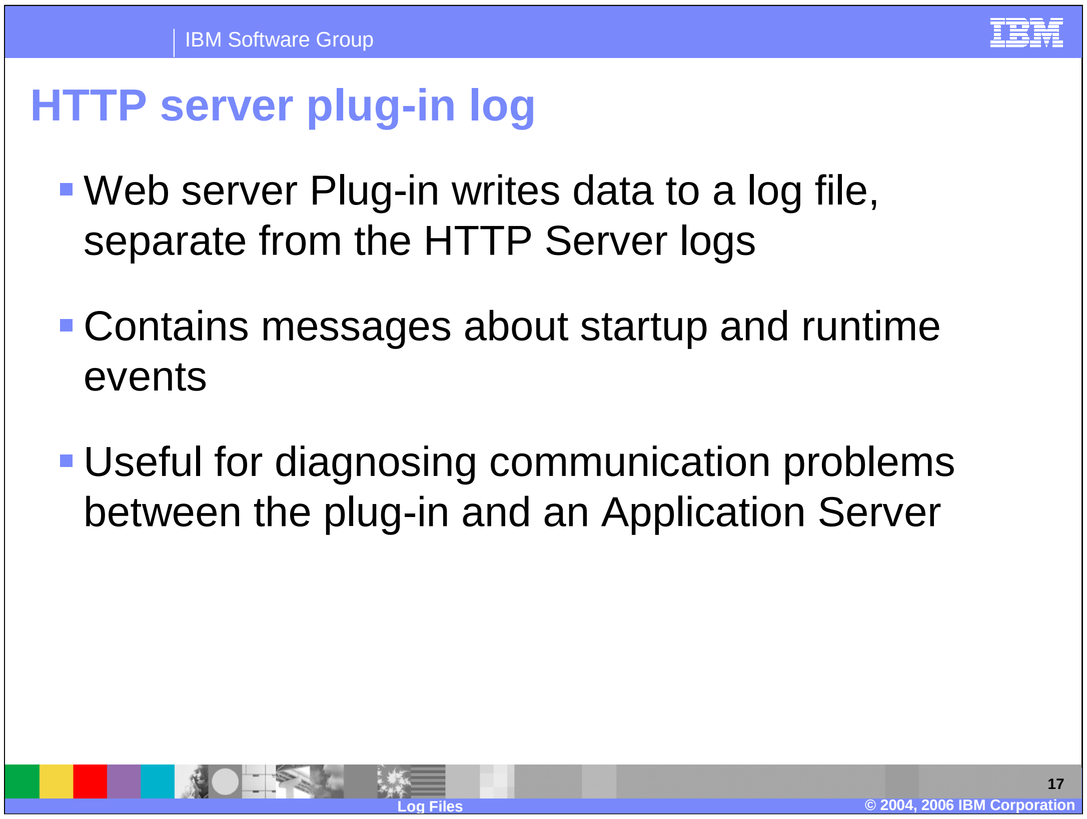
☑ Enable error logging

* Error log file path: ${SERVER_LOG_ROOT}/http_err

* Error log maximum size: 500 MB

* Error log level: Error

The embedded HTTP server logs can be enabled by clicking the link titled "HTTP Error and NCSA Access Logging" from the Application Server's main page in the Administrative Console. To enable the access or error log, you must check the "Enable service at server startup" checkbox, and also the checkbox for the specific log you want to enable. Notice there is no runtime tab for these logs. Logging will only begin once you have saved these changes to your configuration and restarted the Application Server.



HTTP server plug-in log

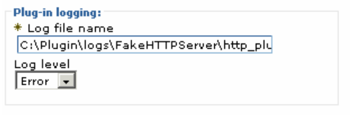
- Web server Plug-in writes data to a log file, separate from the HTTP Server logs
- Contains messages about startup and runtime events
- Useful for diagnosing communication problems between the plug-in and an Application Server

Although it runs inside the web server process, the plug-in writes to its own log file.

The log contains messages about the plug-in's startup process and runtime events. If your web server does not start, you should look for errors in this file, and in the web server's log files. If there seems to be a communication problem between the plug-in and an Application Server, you should also examine this file.

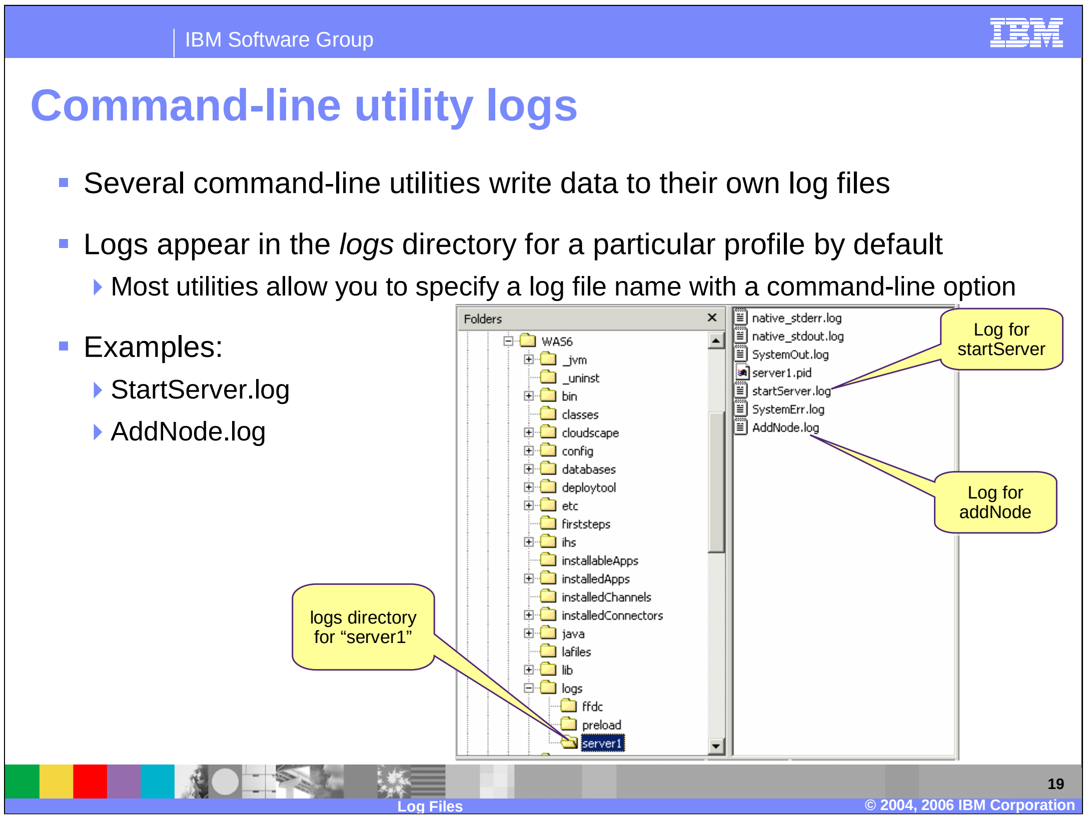
HTTP server plug-in log: Configuration

- Can be set to one of three levels: Error, Warn, or Trace
- Default is Error
- Trace has high overhead
- Configure using the properties page for your Web server in the Administrative Console


Plug-in logging:
Log file name
C:\Plugin\logs\FakeHTTPServer\http_plu
Log level
Error

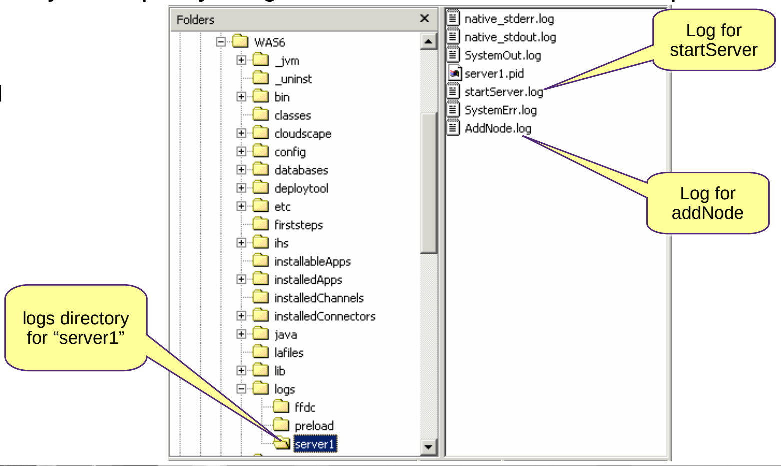
There are three logging levels: Error, Warn, and Trace. The default is Error, which only logs error events. Trace logs every action during plug-in execution. This level of detail is useful for problem determination and is also a great educational resource, because you can see how the plug-in makes decisions. Setting the log level to trace is resource intensive and creates a lot of data in a short time period, so it is not advised to use this level in production unless it is absolutely necessary.

The log level and the location of the log file can be configured in the plug-in configuration file using the Log Element. Remember that the log file will be created on the machine where the web server is running.

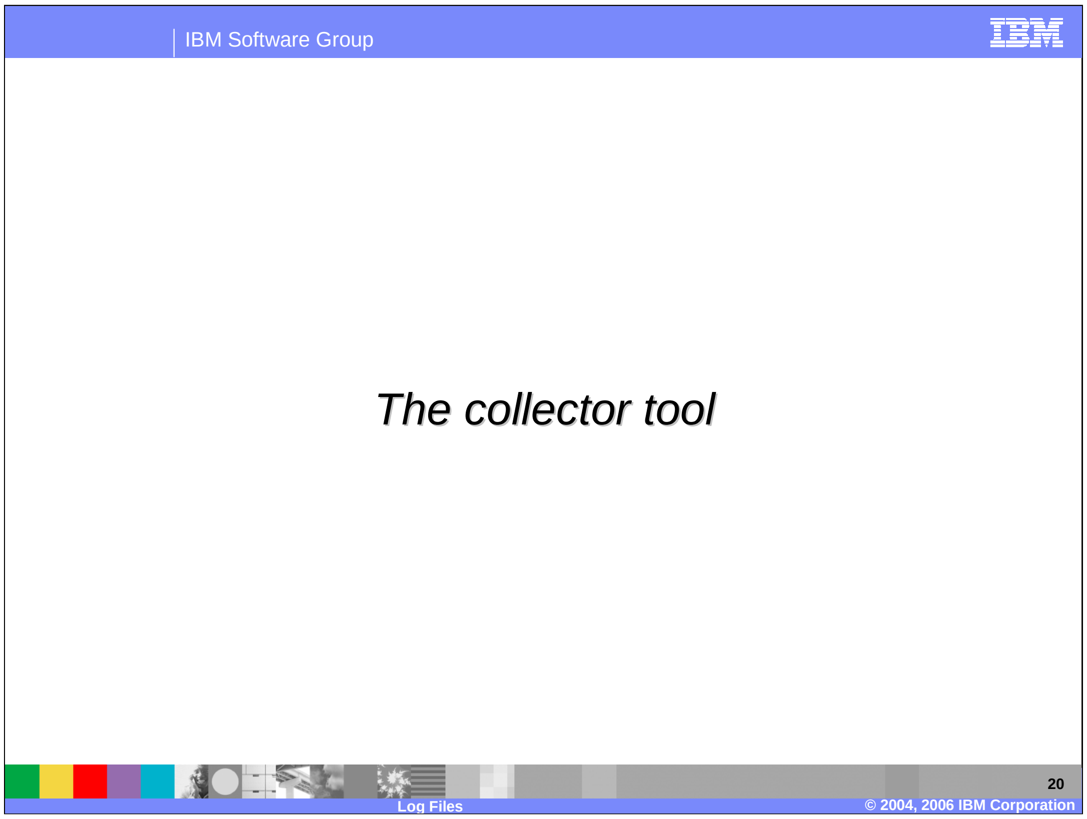


Command-line utility logs

- Several command-line utilities write data to their own log files
- Logs appear in the *logs* directory for a particular profile by default
  - Most utilities allow you to specify a log file name with a command-line option
- Examples:
  - StartServer.log
  - AddNode.log

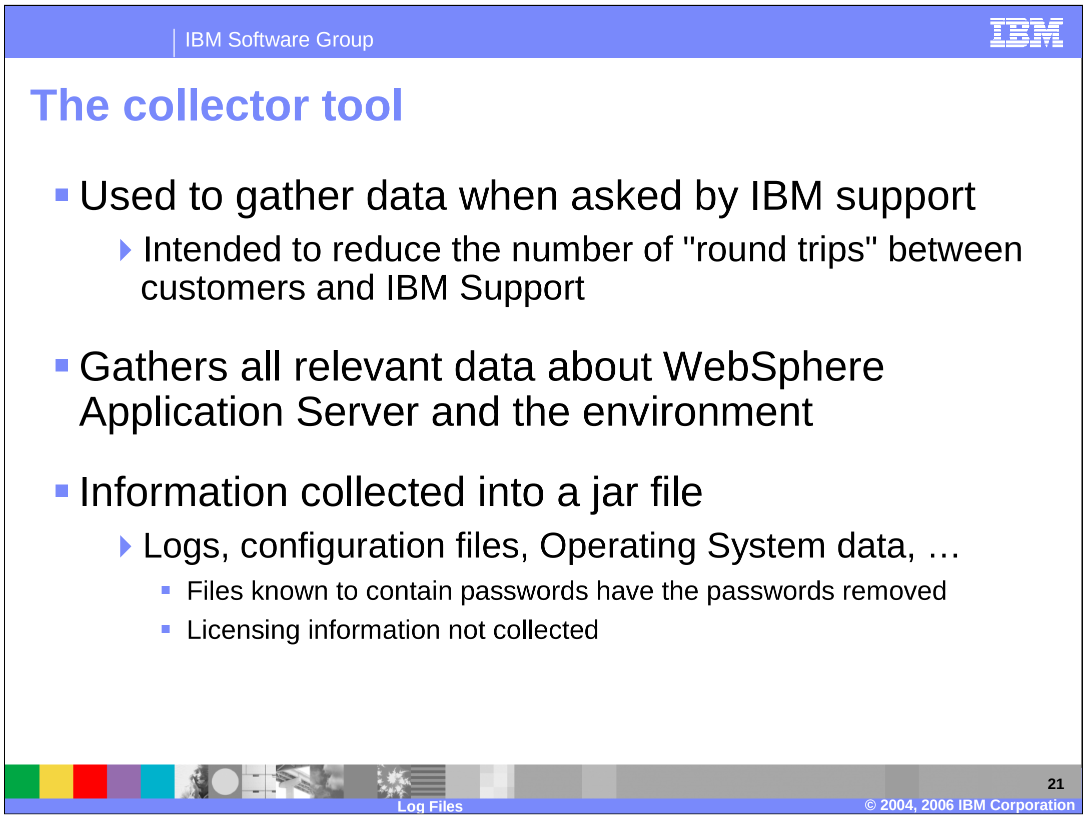


Most of the command-line utilities, such as StartServer or AddNode, write data to their own log files. These logs will appear in your profile’s “logs” directory, and have the same name as the utility. For most utilities, you can specify an alternate log file on the command line if you prefer.



The collector tool

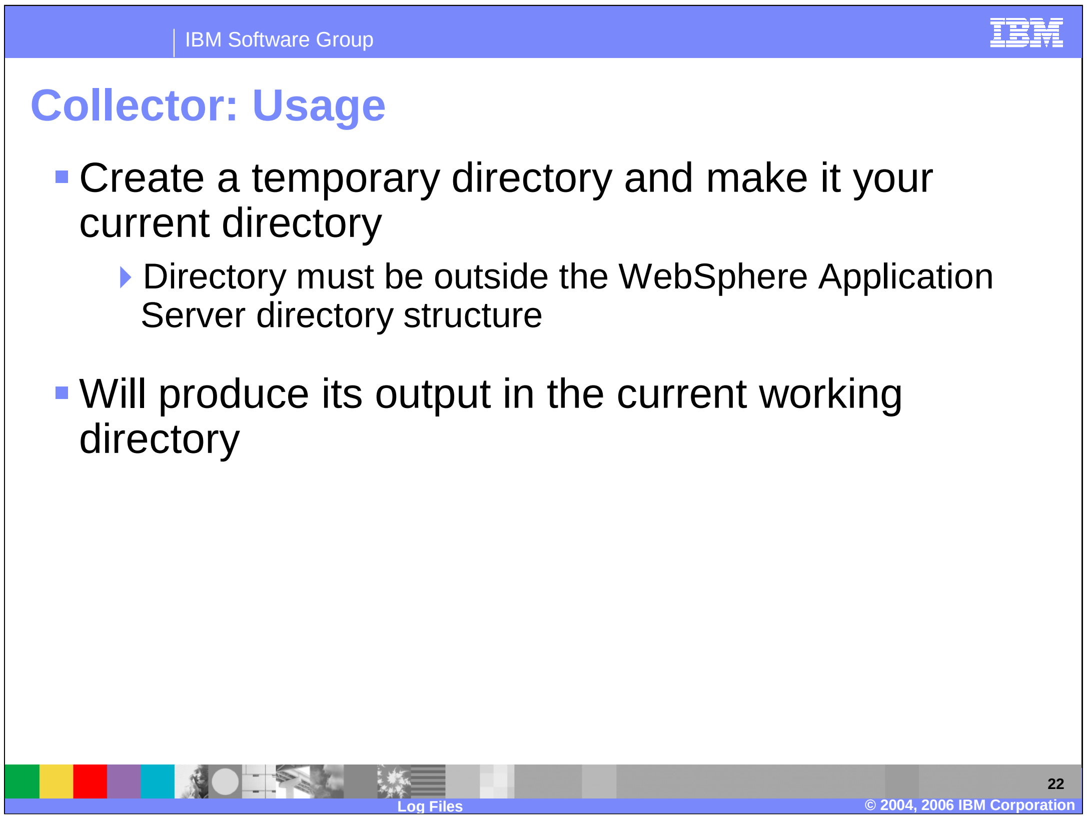
This section will discuss the collector tool.



The collector tool

- Used to gather data when asked by IBM support
  - Intended to reduce the number of "round trips" between customers and IBM Support
- Gathers all relevant data about WebSphere Application Server and the environment
- Information collected into a jar file
  - Logs, configuration files, Operating System data, …
    - Files known to contain passwords have the passwords removed
    - Licensing information not collected

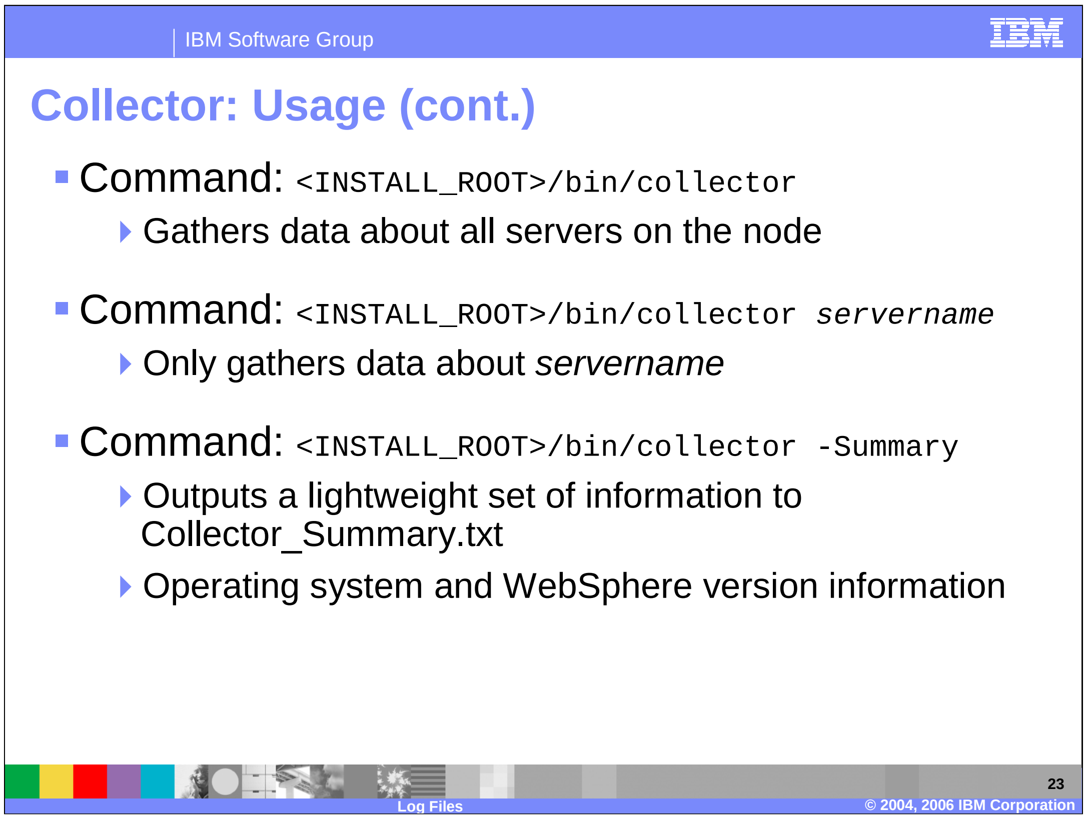
The Collector tool gathers all of the logs and other information necessary for problem determination. It is primarily intended to streamline the interaction between WebSphere Application Server customers and IBM Support. Instead of sending a few files at a time, as the support rep asks for them, you can run this command once and generate a jar file that contains all of the information that might be useful to the support rep, including all of your logs and the configuration repository.



Collector: Usage

- Create a temporary directory and make it your current directory
  - Directory must be outside the WebSphere Application Server directory structure
- Will produce its output in the current working directory

For best results, create a temporary directory outside the WebSphere directory structure, and run the Collector tool from within it to avoid potential file name conflicts.



IBM Software Group

Collector: Usage (cont.)

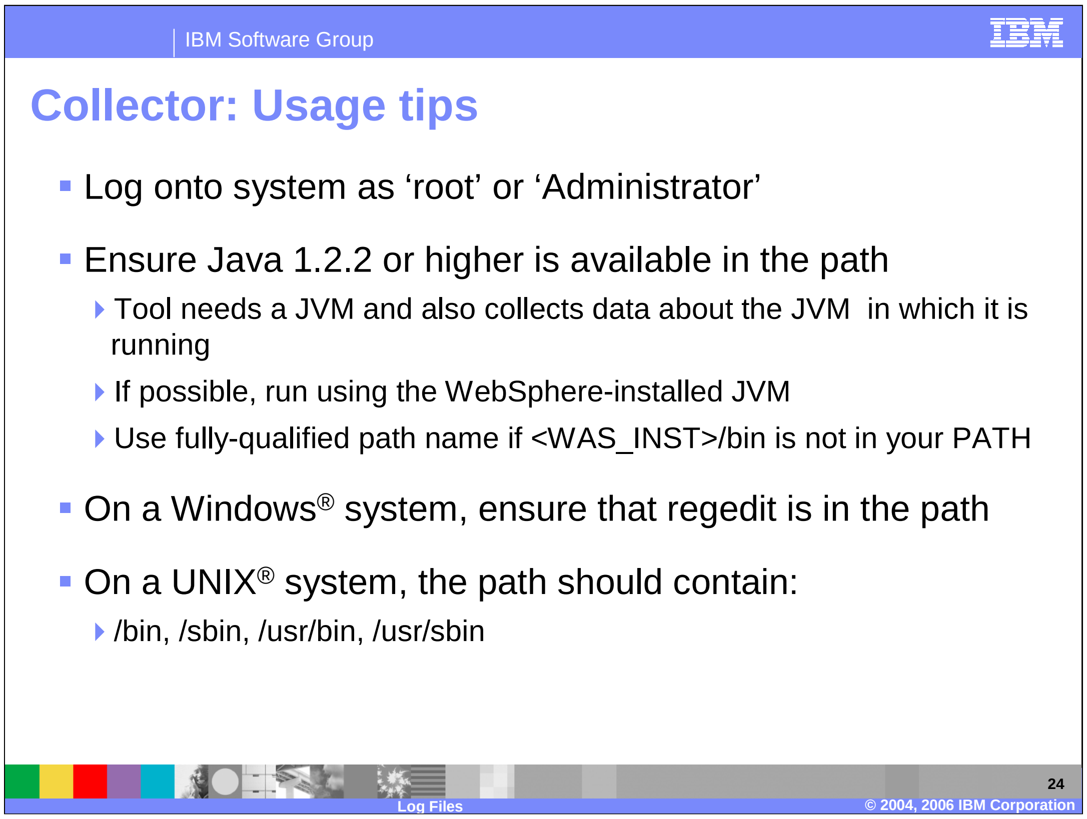
- Command: `<INSTALL_ROOT>/bin/collector`
  - Gathers data about all servers on the node
- Command: `<INSTALL_ROOT>/bin/collector` *`servername`*
  - Only gathers data about *servername*
- Command: `<INSTALL_ROOT>/bin/collector -Summary`
  - Outputs a lightweight set of information to Collector_Summary.txt
  - Operating system and WebSphere version information

23

Log Files

© 2004, 2006 IBM Corporation

The most common use of the tool is to type “<INSTALL_ROOT>/bin/collector”. The tool will create a file called Collector.jar that contains information about every server on your node. You can also specify a server name to collect data for only that server. The summary option is useful for quickly gathering basic information about your machine and WebSphere installation.



IBM Software Group

Collector: Usage tips

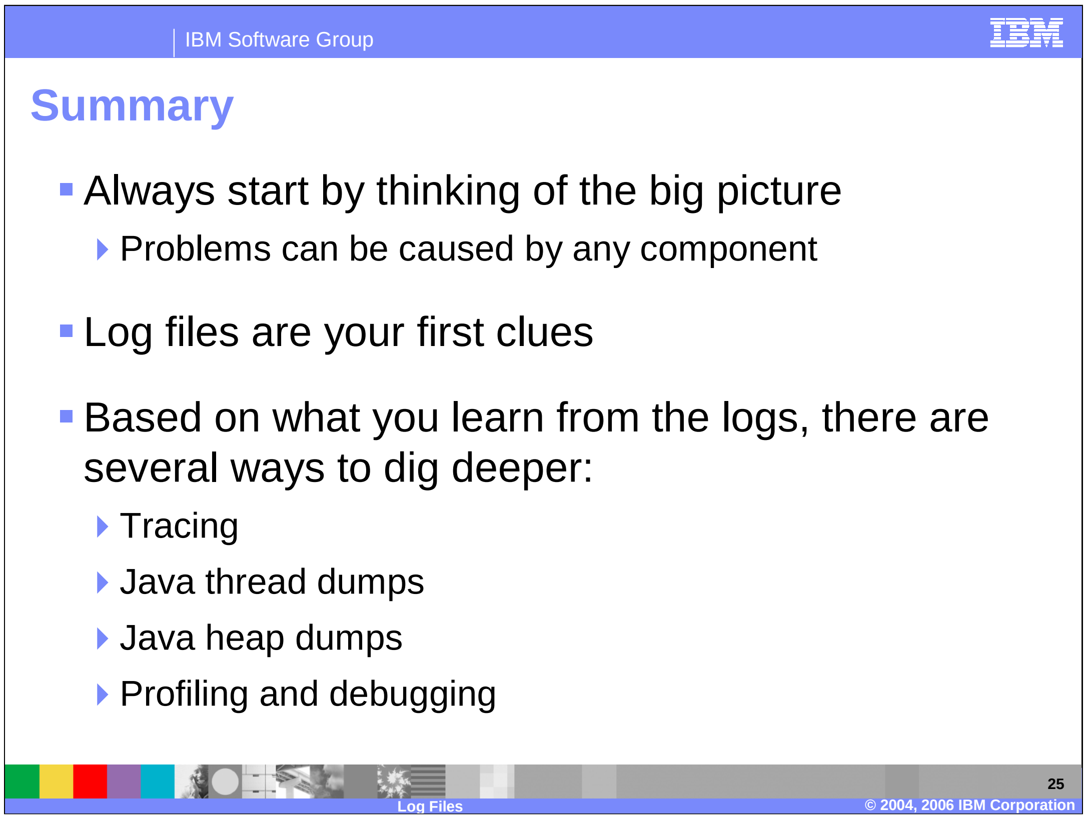
- Log onto system as 'root' or 'Administrator'
- Ensure Java 1.2.2 or higher is available in the path
  - Tool needs a JVM and also collects data about the JVM in which it is running
  - If possible, run using the WebSphere-installed JVM
  - Use fully-qualified path name if <WAS_INST>/bin is not in your PATH
- On a Windows® system, ensure that regedit is in the path
- On a UNIX® system, the path should contain:
  - /bin, /sbin, /usr/bin, /usr/sbin

24

Log Files

© 2004, 2006 IBM Corporation

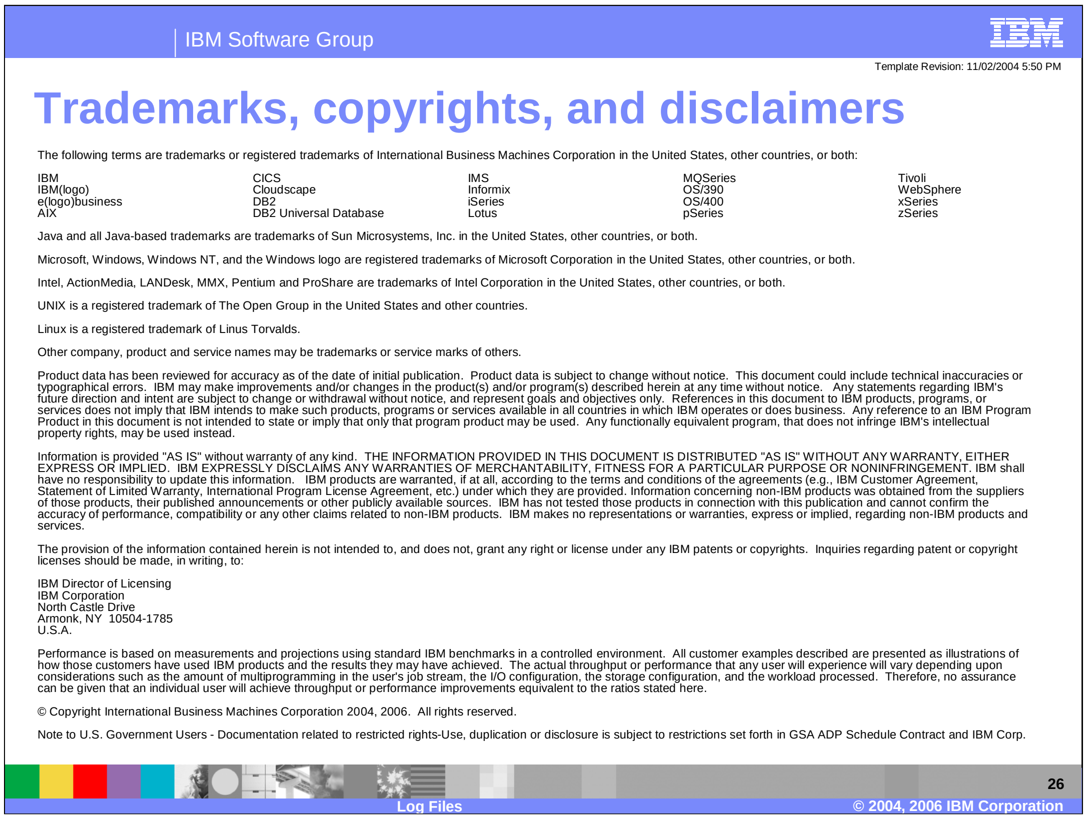
There are a few things you can do to ensure that collector runs smoothly. Run as 'root' or 'administrator' to make sure the tool can access all the necessary data. Also, you'll need to have Java version 1.2.2 or later in your path. On Windows®, make sure that the "regedit" command is in your path, and on UNIX® and Linux® platforms, ensure that '/bin', '/sbin', '/usr/bin', and '/usr/sbin' are in your path.



Summary

- Always start by thinking of the big picture
  - Problems can be caused by any component
- Log files are your first clues
- Based on what you learn from the logs, there are several ways to dig deeper:
  - Tracing
  - Java thread dumps
  - Java heap dumps
  - Profiling and debugging

In summary, this presentation has tried to give you an overview of problem determination tactics for WebSphere Application Server. Always remember to look at the big picture, and consider that any component could be the cause of your problem. Start troubleshooting by looking at the relevant log files. The log files should get you started, and you can dig deeper by employing the tactics discussed in this presentation. Tracing logs detailed information about the execution of WebSphere Application Server. Java thread dumps show you the state of every thread in the Java runtime at a given time. Java heap dumps are a snapshot of the Java memory space at a point in time. Profiling and debugging are methods by which you can use external tools to learn more about the execution of your code and the server.



Trademarks, copyrights, and disclaimers