Log Files
Goals

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

- Look at logs, starting with the most likely component
- Review logs from other components that directly interact with it
# WebSphere logs: Overview

<table>
<thead>
<tr>
<th>Group</th>
<th>Log Files (default names)</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service log</td>
<td>activity.log</td>
<td>Binary log file contains data from each JVM in a node, for analysis using Log Analyzer</td>
</tr>
<tr>
<td>JVM logs</td>
<td><strong>SystemOut.log, SystemErr.log</strong></td>
<td>Contains all messages sent to the Java System.out and System.err streams. These are your main runtime log files.</td>
</tr>
<tr>
<td>Native process logs</td>
<td>native_stdout.log, native_stderr.log</td>
<td>Contain messages sent to stdout and stderr from native code, including the JVM. Mainly used for verbose gc logging.</td>
</tr>
<tr>
<td>Embedded HTTP server logs</td>
<td>http_access.log, http_error.log</td>
<td>Contain all requests to the embedded HTTP server.</td>
</tr>
<tr>
<td>HTTP server plug-in log</td>
<td>http_plugin.log</td>
<td>Contains data about the operation of the HTTP server plug-in module.</td>
</tr>
<tr>
<td>Command-line program logs</td>
<td>startServer.log, addNode.log, &lt;command&gt;.log</td>
<td>Contains data about the execution of individual command-line utilities</td>
</tr>
<tr>
<td>System application &amp; sample application logs</td>
<td>&lt;name&gt;_deploy.txt, &lt;name&gt;_config.txt</td>
<td>Deployment and configuration logs for each of the enterprise applications installed by the WAS installer.</td>
</tr>
</tbody>
</table>
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
- 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.
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.
Configuring JVM logs

- From Servers > Application Servers > servername:
- Logging and Tracing > JVM Logs
- SystemOut and SystemErr logs configured from here
- Logs are self-managing
  - Can roll over based on time or file size
  - Number of historical log files is configurable
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.messaging.version on your Application Server, with a value of “6”
JVM logs: Event types

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>F</td>
<td>Fatal Message</td>
<td>E</td>
<td>Error Message</td>
</tr>
<tr>
<td>W</td>
<td>Warning Message</td>
<td>A</td>
<td>Audit Message</td>
</tr>
<tr>
<td>I</td>
<td>Informational Message</td>
<td>C</td>
<td>Configuration Message</td>
</tr>
<tr>
<td>D</td>
<td>Detail Message</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example Lotus Connections log entries

  com.ibm.lotus.connections.search.searchHandler.GroupLookup warn CLFRW0134W: An error occurred while retrieving wiki groups for user E0EAF6F7-DAE7-4780-98B7-5A8A8C72AD70 from the wikis server.

- [11/17/09 18:01:00:677 EST] 00000021 IndexingNotif I CLFRW0042I: Lotus Connections indexing task 15min-search-indexing-task fired event TaskNotificationInfo.FIRING


Native process logs

- Native logs contain stderr and stdout messages from native code
- Log files do not rollover by default
  - JVM command line parameters can be used to control rollover behavior
    - `-Xverbosegclog[:<file>[,<X>,<Y>]]`
- Filenames are configurable under the “Process Logs” link in the Application Server’s “Logging and Tracing” panel
  - WAS 6.1 → Diagnostic Guide 5.0
  - WAS 7.0 → Diagnostic Guide 6.0
Looking for signs of trouble in native_stderr.log

- If you see these lines it means your JVM has crashed
  - (I)DUMP0007 JVM Requesting System Dump using 'D:\core.20040817.131302.2168.dmp'
  - (I)DUMP0010 System Dump written to D:\core.20040817.131302.2168.dmp

- If you see these lines it means your JVM requested a Java Dump (aka javacore). This does not necessarily represent a crash. It could also be done manually for diagnosing hangs or performance problems.
  - (I)DUMP0007 JVM Requesting Java Dump using 'D:\javacore.20040817.131319.2168.txt'
  - (I)DUMP0010 Java Dump written to D:\javacore.20040817.131319.2168.txt
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_root>/logs by default
- Viewable with Log Analyzer or the “showlog” command
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
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
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
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
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
HTTP server plug-in log output


[Thu Mar 04 15:05:27 2010] 00001210 00000684 - DEBUG: ws_common: websphereUriMatch: Found a match '/homepage' to '/homepage' in UriGroup: default_host_HomepageCluster_URIs with score 9, exact match 9

[Thu Mar 04 15:05:27 2010] 00001210 00000684 - TRACE: ws_common: websphereUriMatch: uri length smaller than uri defs so skipping the rest

[Thu Mar 04 15:05:27 2010] 00001210 00000684 - TRACE: mod_was_ap20_http: as_translate_name: WebSphere will handle: /homepage
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
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
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
Collector: Usage (cont.)

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

Deployment Manager

Web Server
- Plugin
  - Web server logs
  - http_plugin.log
  - logs/server1

Node Agent
- activity.log
- logs/nodeagent

Server 1
- logs/server1

Server 2
- logs/server2

Database logs

Log Output
- logs/dmgr
- logs/nodeagent
- SystemOut.log
- SystemErr.log
- native_stdout.log
- native_stderr.log

Java log files:
- SystemOut.log
- SystemErr.log
- native_stdout.log
- native_stderr.log

Data Flow

© 2010 IBM Corporation
References

This presentation along with several others related to problem determination can be viewed at the following link:

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/index.jsp?topic=/com.ibm.iea.was_v6/was/6.1/ProblemDetermination.html