

Diagnosing My webMethods Server

Version 10.15

October 2022

This document applies to My webMethods Server 10.15 and to all subsequent releases.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Copyright © 2004-2022 Software AG, Darmstadt, Germany and/or Software AG USA, Inc., Reston, VA, USA, and/or its subsidiaries and/or its affiliates and/or their licensors.

The name Software AG and all Software AG product names are either trademarks or registered trademarks of Software AG and/or Software AG USA Inc. and/or its subsidiaries and/or its affiliates and/or their licensors. Other company and product names mentioned herein may be trademarks of their respective owners.

Detailed information on trademarks and patents owned by Software AG and/or its subsidiaries is located at <https://softwareag.com/licenses/>.

Use of this software is subject to adherence to Software AG's licensing conditions and terms. These terms are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

This software may include portions of third-party products. For third-party copyright notices, license terms, additional rights or restrictions, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". For certain specific third-party license restrictions, please refer to section E of the Legal Notices available under "License Terms and Conditions for Use of Software AG Products / Copyright and Trademark Notices of Software AG Products". These documents are part of the product documentation, located at <https://softwareag.com/licenses/> and/or in the root installation directory of the licensed product(s).

Document ID: MWS-DMWS-1015-20221015

Table of Contents

About this Guide	5
Document Conventions.....	6
Online Information and Support.....	6
Data Protection.....	7
1 About My webMethods Server Diagnostic Tools	9
Command Line Diagnostic Tools Requirements.....	10
Diagnostic Portlets Requirements.....	10
Listing Installed Fixes.....	11
2 Using My webMethods Server Command Line Tools	13
Command Line Tools Overview.....	14
envcapture.....	14
envdiff.....	17
loganalyzer.....	19
memorydump.....	23
threaddump.....	25
dbintegritycheck.....	28
Log4j.....	31
Java Artifact Mapping in My webMethods Server.....	31
3 Using External Command-Line Diagnostic Tools	33
Overview of External Command-Line Diagnostic Tools.....	34
Using jconsole to Diagnose My webMethods Server.....	34
Using jconsole in Standard Mode.....	35
Using jvisualvm to Diagnose My webMethods Server.....	35
4 Using My webMethods Server Diagnostic Portlets	37
Using the Thread Dump Portlet.....	38
Using the Performance Analysis Portlet.....	39
Using the Performance Statistics Portlet.....	43
Using the Log Analysis Tools.....	46
Using the Memory Monitor Portlet.....	52
5 Troubleshooting My webMethods Server Using Diagnostic Tools	57
Troubleshooting Configuration Issues.....	58
Troubleshooting Environment Issues.....	64
Troubleshooting Performance Issues.....	66
Troubleshooting Log Issues.....	68
A JMX MBean Attributes and Actions	71
Categories under Root.....	72

Categories under Core.....74

About this Guide

■ Document Conventions	6
■ Online Information and Support	6
■ Data Protection	7

This guide describes how to use the My webMethods Server diagnostic tools. It contains information for administrators who want to administer and diagnose My webMethods Server.

To use this guide effectively, you should be familiar with the setup and operation of My webMethods Server.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies service names and locations in the format <i>folder.subfolder.service</i> , APIs, Java classes, methods, properties.
<i>Italic</i>	Identifies: Variables for which you must supply values specific to your own situation or environment. New terms the first time they occur in the text. References to other documentation sources.
Monospace font	Identifies: Text you must type in. Messages displayed by the system. Program code.
{ }	Indicates a set of choices from which you must choose one. Type only the information inside the curly braces. Do not type the { } symbols.
	Separates two mutually exclusive choices in a syntax line. Type one of these choices. Do not type the symbol.
[]	Indicates one or more options. Type only the information inside the square brackets. Do not type the [] symbols.
...	Indicates that you can type multiple options of the same type. Type only the information. Do not type the ellipsis (...).

Online Information and Support

Product Documentation

You can find the product documentation on our documentation website at <https://documentation.softwareag.com>.

In addition, you can also access the cloud product documentation via <https://www.softwareag.cloud>. Navigate to the desired product and then, depending on your solution, go to “Developer Center”, “User Center” or “Documentation”.

Product Training

You can find helpful product training material on our Learning Portal at <https://knowledge.softwareag.com>.

Tech Community

You can collaborate with Software AG experts on our Tech Community website at <https://techcommunity.softwareag.com>. From here you can, for example:

- Browse through our vast knowledge base.
- Ask questions and find answers in our discussion forums.
- Get the latest Software AG news and announcements.
- Explore our communities.
- Go to our public GitHub and Docker repositories at <https://github.com/softwareag> and <https://hub.docker.com/publishers/softwareag> and discover additional Software AG resources.

Product Support

Support for Software AG products is provided to licensed customers via our Empower Portal at <https://empower.softwareag.com>. Many services on this portal require that you have an account. If you do not yet have one, you can request it at <https://empower.softwareag.com/register>. Once you have an account, you can, for example:

- Download products, updates and fixes.
- Search the Knowledge Center for technical information and tips.
- Subscribe to early warnings and critical alerts.
- Open and update support incidents.
- Add product feature requests.

Data Protection

Software AG products provide functionality with respect to processing of personal data according to the EU General Data Protection Regulation (GDPR). Where applicable, appropriate steps are documented in the respective administration documentation.

1 About My webMethods Server Diagnostic Tools

■ Command Line Diagnostic Tools Requirements	10
■ Diagnostic Portlets Requirements	10
■ Listing Installed Fixes	11

Command Line Diagnostic Tools Requirements

The command line diagnostic tools for My webMethods Server require Java Runtime Environment version 1.8 (JRE 1.8) or later. When you install My webMethods Server Diagnostic Tools, the Software AG Installer automatically installs and configures the required JRE.

You can use the command line tools to diagnose earlier versions of My webMethods Server (specifically, versions 8.2.x, 8.0.x, and 7.1.x). To do so, you must:

- Configure the corresponding server installation to use JRE 1.8 or later.
- Make sure the release installation is updated to the following level:
 - For a version 7.1.x installation, MWS_7.1.3_Fix 5 or later.
 - For a version 8.0.x installation, MWS_8.0_SP2_Fix4 or later.

You can download fix packages for My webMethods Server from the Software AG Empower Product Support Website, <http://empower.softwareag.com> (login required).

- Start the server in monitoring mode using one of the following commands:

- Windows:

```
Software AG_directory\MWS\bin\mws.bat -s server_name -m start
```

```
Software AG_directory\MWS\bin\mws.bat -s server_name -m run
```

- UNIX:

```
Software AG_directory\MWS\bin\mws.sh -s server_name -m start
```

```
Software AG_directory\MWS\bin\mws.sh -s server_name -m run
```

For more information, see the topic “Command Syntax for My webMethods Server” in the PDF guide *Administering My webMethods Server*.

Diagnostic Portlets Requirements

My webMethods Server diagnostic portlets are available for use only with My webMethods Server 8.2 or later. You cannot use these diagnostic portlets to diagnose My webMethods Server 8.0.x or earlier.

However, you can use the diagnostic portlets that run on My webMethods Server 8.2 to analyze `envcapture.zip` files generated by diagnostic command line tools running on 7.1.x or My webMethods Server 8.0.x. For more information, see:

- [“envcapture” on page 14](#)
- [“Using the Log Analysis Tools” on page 46](#)

Listing Installed Fixes

You can determine the fixes that have been applied to My webMethods Server using the following methods:

- Review the `_full_.log` file.
- Execute the `envcapture` command.
- Execute the `loganalyzer` command to view fixes associated components that have errors.

Reviewing the `_full_.log` File

My webMethods Server determines the installed fixes at server startup and logs the list of installed fixes to the `_full_.log` file. My webMethods Server `_full_.log` files reside in the following directory:

My webMethods Server_directory \server\server_instance\log

For more information about this log, see *Administering My webMethods Server*.

Because determining the list of fixes can extend the start-up time, it is possible to disable this action at server startup. To do so, see the topic "Preventing My webMethods Server from Listing Installed Fixes at Startup" in the PDF publication *Administering My webMethods Server*.

Using the `envcapture` Command to List Installed Fixes

The `envcapture` command provides environment information for My webMethods Server, such as information about log files, information about configuration files, and details about the local operating system and Java virtual machine. The `envcapture` command also provides a list of fixes that have been applied to My webMethods Server. For more information, see [“envcapture” on page 14](#).

Using the `loganalyzer` Command to List Installed Fixes Associated with Components with Errors

The `loganalyzer` command analyzes reported issues in log files, for example, the `errors.log` file. If you specify that you want `loganalyzer` to provide component-oriented output that displays the components that cause the events in the log, the `loganalyzer` command also lists fixes, if any, that are applied to the components.

2 Using My webMethods Server Command Line Tools

■ Command Line Tools Overview	14
■ envcapture	14
■ envdiff	17
■ loganalyzer	19
■ memorydump	23
■ threaddump	25
■ dbintegritycheck	28
■ Log4j	31
■ Java Artifact Mapping in My webMethods Server	31

Command Line Tools Overview

You can start or stop My webMethods Server from the command line. When you need to identify and resolve issues, start My webMethods Server in monitoring mode. For more information, see [“Command Line Diagnostic Tools Requirements” on page 10](#). Review the usage notes to determine if the My webMethods Server command line tool requires additional permissions.

envcapture

Description

Creates an environment-specific report of the My webMethods Server environment that you can view or save. If you want review the server's environment later on, you can create a current report and compare it to a previous report.

You would typically run the envcapture command before and after environment changes such as updates, upgrades, configuration changes, or component deployments to the server environment. For on-premises installations, the system runs envcapture automatically to capture server environment information after you install or upgrade My webMethods Server.

The envcapture command captures the following information:

- Log files.
- Configuration files.
- Applied fixes.
- Installation folder structure, size, time stamps, and file system permissions for the current operating system user account.
- Local operating system and Java virtual machine (JVM) information.
- Internet connectivity and configuration settings, and the connectivity to My webMethods Server database information.
- Connectivity information from the database connection configuration file, mws.db.xml.

The envcapture command uses the standard `System.getProperties()` and `System.getenv()` methods to obtain the system specific information. When you execute the envcapture command without specifying the `-s | --server` option, it captures information about the default My webMethods Server instance. All information is saved in an envcapture.zip file.

The envcapture command-line tool is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin.

Syntax

- On Windows: `envcapture.bat rootDir [options]`

- For UNIX: `./envcapture rootDir [options]`

Options

```
[{-d | --mws-db-xml} path_to_db_configuration_file]
[{-a | --action} {all | filesystem | network | env | java | db | logs}]
[{-f | --from-date} {yyyy.MM.dd | HH:mm:ss | nd | nH}]
[{-t | --to-date} {yyyy.MM.dd | HH:mm:ss | nd | nH}]
[{-s | --server} server_instance]
[{-o | --output} directory_path/name]
```

The following table lists the arguments and options of the envcapture command-line tool:

Argument or Option	Description
<code>rootDir</code>	Required. Specifies the directory path of the My webMethods Server instance.
<code>{-d --mws-db-xml}</code>	Specifies the directory path to the mws.db.xml configuration file for the installed database; provide the fully qualified path name as the argument.
<code>{-a --action}</code>	Specifies the information to capture. The valid arguments are as follows: <ul style="list-style-type: none"> ■ <code>all</code> queries and captures all of the environment information; <code>all</code> is the default value. ■ <code>filesystem</code> queries and captures information about the file system structure. ■ <code>network</code> queries and captures information about the network and Internet connection. ■ <code>env</code> queries and captures information about the environment variables. ■ <code>java</code> queries and captures information about the Java system properties. ■ <code>db</code> queries and captures information about the installed database. ■ <code>logs</code> queries and captures logging information.
<code>{-f --from-date}</code>	Specifies the start date to use for capturing logging information using the <code>year.month.day</code> or <code>hour.minute.second</code> date format as follows: <code>yyyy.MM.dd</code> or <code>HH:mm:ss</code> . You can also specify an interval as a number of days (d) or hours (h), for example enter 2 days as <code>2d</code> , or 3 hours as <code>3h</code> . This captures information from the logs that occurred within the specified time period (for example, collect the logs that have been modified in the last 3 hours).

Argument or Option	Description
{-t --to-date}	Specifies the end date to use when capturing logging information, using the same date format is the same as the from date option. You can also input a time interval using hours and days using, for example, 2d or 3h. This captures information from the logs that occurred prior to the specified date or interval (for example, collect the logs that have been modified up until 3 hours ago).
{-s --server}	Runs the envcapture command to capture information about another server instance available on the local system. If you do not specify a server instance, the tool performs environment capture on the default My webMethods Server instance.
{-o --output}	Stores the result of the environment capture in an alternative location. By default, the command creates the result files in the same directory in which you started it.
{-h --help}	Displays command help text about the command, the available options and arguments, and default values.

Usage Notes

- If your Internet connection uses proxy settings, as a prerequisite you must configure the corresponding proxy host and port for your My webMethods Server in the Windows `server.properties.bat` file which is located in the *Software AG_directory* `/MWS/server/server_name/bin` directory. When working on UNIX-based operating systems, update the `server.properties.sh` file. The envcapture command verifies the settings and the Internet access of the server.
- The envcapture command produces the result using the log4j mechanism. For more information, see [“Log4j” on page 31](#).
- To use the envcapture output to compare before and after server environment changes, see [“envdiff” on page 17](#).

Examples

- Windows:

```
envcapture.bat C:\softwareag\MWS\ -d C:\DB\mws.db.xml -a all  
-s myServer -o C:\softwareag\MWS\tools\diagnostics\bin\output\
```

- UNIX:

```
./envcapture /opt/softwareag/MWS/ -d /opt/DB/mws.db.xml -a all  
-s myServer -o /opt/softwareag/MWS/tools/diagnostics/bin/output/
```

envdiff

Description

Compares the XML file results from two separate output files generated by the envcapture command-line tool and helps you identify issues related to environment changes. For more information, see [“envcapture” on page 14](#).

You can run the envdiff command on your local instance of My webMethods Server to obtain a baseline reference file. At a later time, you can run the envcapture command again and then compare the two files with the envdiff command to examine the following:

- Folder structure
- File size
- MD5 hash
- Timestamps
- Permissions

The envdiff command options enable you to manually exclude artifacts such as a specific directory's sub-trees from the comparison.

In addition, you can display the results as a plain text report in the console window, or save them as a merged XML file. The merged file outlines the differences and identifies the originating documents.

The envdiff command is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin.

Syntax

- On Windows: `envdiff.bat file_A file_B [options]`
- On UNIX: `./envdiff.bat file_A file_B [options]`

Options

```
[{-r | --ignore-attributes} {size | md5Hash | timestamp | permissions}]
[{-u | --ignore-subtrees} directory_name]
[{-x | --xml}]
[{-o | --output} directory_path/filename]
```

The following table lists the arguments and options that the envdiff command-line tool accepts:

Argument or Option	Description
<i>file_A</i>	The path and file name of the first of the two files you want to compare.
<i>file_B</i>	The path and file name of the second of the two files you want to compare.
{-r --ignore-attributes}	Specifies the attributes to exclude from the file comparison. To exclude multiple attributes, separate the items using a comma. Valid attributes for this option are as follows: <ul style="list-style-type: none">■ size■ md5Hash■ timestamp■ permissions
{-u --ignore-subtrees}	Specifies the directory sub-trees to exclude from the comparison. To exclude multiple directories, separate the items using a comma.
{-x --xml}	Displays XML output in the console window.
{-o --output}	Saves the XML output to a file on the file system and does not display it in the console window. Valid only with the {-x --xml} option.
{-h --help}	Displays command help text about the command, the available options and arguments, and default values.

Usage Notes

After comparing the files, the results show the following:

- Expected and actual size of the directories; the size of the directories could differ if there are changes in the directory structure, for example, new directories are created in the structure.
- Differences in a specific file by comparing the content of the file, based on the captured md5Hash information.
- Expected and actual size of files and the corresponding time stamps.
- Differences resulting from new, updated and deleted files or directories.
- Differences in user permissions of the directories.

Examples

- Windows:

```
envdiff.bat C:\Compare_XMLs\reference.xml
           C:\Compare_XMLs\current.xml
           -r size,md5Hash -u temp,dir1 -x -o C:\Compare_XMLs\Result\
```

- UNIX:

```
./envdiff /opt/Compare_XMLs/reference.xml
          /opt/Compare_XMLs/current.xml
          -r size,md5Hash -u temp,dir1 -x -o /opt/Compare_XMLs/Result/
```

loganalyzer

Description

Identifies and analyzes the reported issues found in XML log files, then organizes the results into reports displayed in the console window. The My webMethods Server loganalyzer command stores the results as plain text in the diagnostics.log file.

You can set loganalyzer command options to produce the following reports file:

- **productEvent.xml**. Reports the name of products that caused events reports in the logs.
- **loggerEvent.xml**. Reports the names of components that caused the event reports in the logs.
- **logLevel.xml**. Reports a summary of all available events found in the logs.
- **messageBased.xml**. Reports a summary of the issues with common message texts found in the logs.
- **componentEvent.xml**. Reports the Java artifacts such as jars, portlets, or web applications that caused the events found in the log.

To correctly identify the component or product causing the error, the loganalyzer command uses the stack trace, when available, or the logger element. The stack trace and logger element are part of the issue details reported with fatal issues in the errors.log.

The loganalyzer command parses the stack trace or logger to find the corresponding component and product mapped in the javaArtifactMapping.xml file. For more information, see [“Java Artifact Mapping in My webMethods Server” on page 31](#).

The loganalyzer command is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

Syntax

- On Windows: `loganalyzer.bat logFile [options]`
- On UNIX: `./loganalyzer logFile[options]`

Options

```
[{-f | --from} {yyyy.MM.dd HH:mm:ss | nd | nH}]
```

```
[{-t | --from-date} {yyyy.MM.dd
HH:mm:ss | nd | nH}]
[{-i | --similarity} integer]
[{-j | --javaArtifactMapping} directory_and_file_name]
[{-s | --server} server_instance]
[{-a | --action} {all | frequency | severity | component | product
| logger}]
[{-e | --severity} {DEBUG | ERROR | INFO | FATAL | WARN }]
[{-x | --xml}]
[{-o | --output} directory_name]
```

The following table lists the arguments and options of the loganalyzer command-line tool:

Arguments and Options	Description
<i>logfile</i>	<p>Specifies the full directory path and name of the log file to analyze, or specifies a directory that contains the errors.log file.</p> <p>When you provide a directory without specifying a filename, the loganalyzer command searches for a log file by checking the subdirectories for file names matching the following:</p> <ul style="list-style-type: none"> ■ errors.log ■ errors.xml ■ errors <p>The loganalyzer command scans the first file matching the file name criteria for warn, error, and fatal events that are generated by My webMethods Server.</p>
{-f --from-date}	<p>Specifies the date from which to start analyzing the selected log file. Use <i>yyyy.MM.dd HH:mm:ss</i> format for the date or use days (d) or hours (h), for example, 2d or 3h. This captures information from the logs that occurred within the specified time period (or, for example, the logs that have been modified in the last 3 hours). The default value is 1970.01.01 02:00:00.</p>
{-t --to-date}	<p>Specifies the date to stop analyzing the selected log file. Use <i>yyyy.MM.dd HH:mm:ss</i> format for the date or use days (d) or hours (h), for example, 2d or 3h. This captures information from the logs that occurred prior to the specified date (or, for example, logs that have been modified up until 3 hours ago). The default value is 1970.01.01 02:00:00.</p>
{-i --similarity}	<p>Specifies a percentage for the similarity level by which log messages are considered equal or dissimilar.</p>

Arguments and Options	Description
{-j --javaArtifactMapping}	<p data-bbox="727 260 1479 327">Specifies the full path to the directory of the <code>javaArtifactMapping.xml</code> file.</p> <p data-bbox="727 352 1479 453">If you do not specify a location for the <code>javaArtifactMapping.xml</code> file, the loganalyzer searches for the file in the following locations:</p> <ul data-bbox="727 478 1479 961" style="list-style-type: none"> <li data-bbox="727 478 1479 579">■ When you provide only a directory location for the <code>logfile</code> argument, the loganalyzer searches for the <code>javaArtifactMapping.xml</code> file in the specified directory. <li data-bbox="727 604 1479 961">■ When you provide a directory location and file name for the <code>logfile</code> argument, the loganalyzer searches for the <code>javaArtifactMapping.xml</code> file by traversing the directories to find the <code>logfile</code> and then does the same search to find the <code>javaArtifactMapping.xml</code> file. For example, if the <code>logfile</code> is in <code>C:\Directory1\Directory2\Directory3\errors.log</code>, then the expected location and name of the <code>JavaArtifactMapping.xml</code> is <code>C:\Directory1\Directory2\config\javaArtifactMapping.xml</code> file.
{-s --server}	<p data-bbox="727 982 1479 1083">Specifies a server available on the local system. When you do not specify a server instance, the loganalyzer uses the default My webMethods Server instance.</p>
{-a --action}	<p data-bbox="727 1104 1479 1184">Specifies the information type to analyze. The default value is <code>frequency</code>. The valid values for this option are as follows:</p> <ul data-bbox="727 1209 1479 1428" style="list-style-type: none"> <li data-bbox="727 1209 1479 1245">■ <code>all</code> captures information about all messages. <li data-bbox="727 1270 1479 1337">■ <code>frequency</code> captures information about messages by the frequency of their appearance. <li data-bbox="727 1362 1479 1428">■ <code>severity</code> captures information about messages according to their severity. <div data-bbox="776 1444 1461 1612" style="background-color: #f0f0f0; padding: 5px;"> <p data-bbox="776 1461 857 1488">Note:</p> <p data-bbox="776 1493 1461 1596">You can specify the <code>{-e --severity}</code> option to narrow the results in the frequency report to include only the results that have a severity level you specify.</p> </div> <ul data-bbox="727 1629 1479 1881" style="list-style-type: none"> <li data-bbox="727 1629 1479 1696">■ <code>component</code> captures information about environment-related messages. <li data-bbox="727 1722 1479 1789">■ <code>product</code> captures information about Java-related messages <li data-bbox="727 1814 1479 1881">■ <code>logger</code> captures information about database-related messages.

Arguments and Options	Description
<code>{-e --severity}</code>	<p>Reports on the severity level that you supply. You can only use this option when also using the <code>action</code> option with the frequency value. The default value is <code>INFO</code>. Specify one of the following severity levels:</p> <ul style="list-style-type: none">■ <code>DEBUG</code>■ <code>ERROR</code>■ <code>FATAL</code>■ <code>INFO</code>■ <code>WARN</code>
<code>{-x --xml}</code>	Displays XML output in the console window.
<code>{-o --output}</code>	Saves the XML output to a file on the file system and does not display it in the console window. Valid only with the <code>{-x --xml}</code> option.
<code>{-h --help}</code>	Displays command help text about the command, the available options and arguments, and default values.

Usage Notes

- The `loganalyzer` command detects changes resulting from installing a single fix of the product because the version numbers of the internal JAR files and the product components might differ.
- If you specify that you want the `loganalyzer` command to provide component-oriented output that displays the components that cause the events in the log, `loganalyzer` also lists fixes, if any, that are applied to the components.
- Before running the `loganalyzer` command, you might want to update the `javaArtifactMapping.xml` file if you have installed and deployed any custom components on My webMethods Server. To update the `javaArtifactMapping.xml` file, see [“Updating the Java Artifact Mapping File” on page 32](#).
- After running the `loganalyzer` command and determining the product and component raising errors, you might want to narrow your results. You can exclude classes of that component and product from the logging information and the displayed statistics. The next time you run `loganalyzer`, the result shows other components and products from the stack trace that are possible candidates responsible for the error. Use the `excludedMappingClasses.xml` file to list class names to exclude from the query of components and products that cause errors.
- The `loganalyzer` command uses the `errors.log` which is a `log4j`-produced result. For more information, see [“Log4j” on page 31](#).

Examples

- Windows example:

```
loganalyzer.bat C:\softwareag\MWS\tools\diagnostics\bin\errors.log
-f 2d -t 1d -i 90
-j C:\softwareag\MWS\tools\diagnostics\config\javaArtifactMapping.xml
-s myServer -a all -x -o C:\softwareag\MWS\tools\diagnostics\bin\output
```

- UNIX example:

```
./loganalyzer /opt/softwareag/MWS/tools/diagnostics/bin/errors.log
-f 2d -t 1d -i 90
-j /opt/softwareag/MWS/tools/diagnostics/config
  /javaArtifactMapping.xml
-s myServer -a all -x -o /opt/softwareag/MWS/tools/diagnostics
  /bin/output
```

- The following example shows a loganalyzer command that uses the default `{-a | --action}` argument frequency to capture log information by the frequency of appearance. In this case, this creates a frequency report from a log4j XML file produced by the `dbintegritycheck` command. The example command also includes the `--severity ERROR` option to narrow the results so that the frequency report includes only items with severity level ERROR:

```
loganalyzer.bat --severity ERROR
C:\softwareag\MWS\tools\diagnostics\bin\db_integrity_report.xml
```

memorydump

Description

Captures information about memory allocation of My webMethods Server instances that run on local and remote systems. Use the `memorydump` command to identify possible memory leaks or troubleshoot any existing memory allocation problems.

The command creates a `heapdump-yyyy-MM-dd-HH_mm_ss_z.hprof` file after it collects the information. To open and analyze the file, you can use the standard JDK 1.6 `jVisualVM` tool available in your `java/bin` directory or use an alternative Java profiler tool.

You can specify options that gather information from a local or remote system, and the name of the server instance you want to diagnose. When the `memorydump` command creates a large results file, you can set a command option to compress the output and create a smaller archive file, `heapdump-yyyy-MM-dd-HH_mm_ss_z.hprof.zip`. When you use the compressed file option, the compressed file is stored in the file system and the uncompressed copy is deleted.

You cannot compress the result when diagnosing the memory allocation of My webMethods Server on a remote system.

The `memorydump` command is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

Syntax

- On Windows: `memorydump.bat -b username -g password [options]`
- On UNIX: `./memorydump -b username -g password [options]`

Options

```
[{-n | --host} host_name]
[{-p | --port} port_number]
[{-k | --ssl} {true | false}]
[{-c | --compress}]
[{-s | --server} server_name]
[{-o | --output} directory_path/name]
```

The following table lists the arguments and options of the memorydump command-line tool:

Arguments and Options	Description
{-b --username}	Required. Specifies a user name that can connect to the server.
{-g --password}	Required. Specifies a password for the specified user name.
{-n --host}	Provides the host name of My webMethods Server; uses the default value of <code>localhost</code> when no value is specified; supply a <code>host_name</code> when My webMethods Server is running remotely.
{-p --port}	Specifies the number of the JMX monitoring port of the specified server instance. When the port number is not specified, the command uses the default port number, 5002.
{-k --ssl}	Configures communication over SSL. The default value is <code>false</code> . If My webMethods Server is configured to use JMX over SSL, set this value to <code>true</code> .
{-c --compress}	Compresses the file and adds it to an archive (.zip) file. This option does not apply when the command-line tool is run on server instances located on a remote system.
{-s --server}	Executes the command on the specified server instance available on the local system. When unspecified, the default instance is used. Do not use this option to specify a server instance on a remote system. Use the {-p --port} option to specify the JMX port used by the remote system.
{-o --output}	Saves and stores the XML results in the specified location and with a different file name; the results are not displayed in the console window. Required when running the command on a remote system.

Arguments and Options	Description
{-h --help}	Displays command help text for the command, the available options and arguments, and default values.

Usage Notes

- You must specify the full location path to the result including the exact file name.
- You can run the command only on Java Runtime Environment version 1.6 (JRE 1.6) and later. Additionally, you can only run the command on a HotSpot Java virtual machine (JVM) provided by Oracle Corporation. You cannot run the command on other JVMs.

Security Considerations

The memorydump command uses Java Management Extensions (JMX) to establish a remote connection to the server and to obtain the memory dump information. You must provide user credentials in the command using credentials that have My webMethods Server Connect to MWS JMX functional privilege, `mws.privilege.jmx.connect`.

Examples

- Windows example:

```
memorydump.bat -c -s myServer
-o C:\softwareag\MWS\tools\diagnostics\bin\output\fileName.hprof
-b myUserName -g myPassword
```

- UNIX example:

```
./memorydump -c -s myServer
-o opt/softwareag/MWS/tools/diagnostics/bin/output/fileName.hprof
-b myUserName -g myPassword
```

threaddump

Description

Enables monitoring thread execution on My webMethods Server local and remote servers. The threaddump command uses Java Management Extensions (JMX) to establish a remote connection to the server and then obtains the thread dump information.

By default, threaddump is configured to obtain information about the local system running on port 5002. To run the command to obtain information from a remote system, you must configure the `-n | --host` option.

The threaddump command is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

Syntax

- On Windows: `threaddump.bat -b username -g password [options]`
- On UNIX: `./threaddump -b username -g password [options]`

Options

```
[{-a | --action} {all | deadlocked}]
[{-n | --host} host_name]
[{-p | --port} port_number]
[{-e | --stack-depth} integer]
[{-s | --server} server_instance]
[{-b | --username} user_name]
[{-g | --password} password]
[{-k | --ssl} {true | false}]
[{-x | --xml}]
[{-o | --output} directory_path_file_name]
[{-h | --help}]
```

The following table lists the arguments and options of the threaddump command-line tool:

Arguments and Option	Description
{-b --username}	Required. Specifies a user name that can connect to the server.
{-g --password}	Required. Specifies a password for the specified user name.
{-a --action}	Specifies the information type to capture. Specify one of the following: <ul style="list-style-type: none"> ■ <code>all</code> to monitor the status of all threads (default). ■ <code>deadlocked</code> to identify server processes experiencing a thread deadlock.
{-n --host}	Provides the host name of My webMethods Server; uses the default value of <code>localhost</code> when no value is specified; supply a <code>host_name</code> when My webMethods Server is running remotely.
{-p --port}	Specifies the number of the JMX monitoring port. When the port number is not specified, the command uses the default port number, 5002.
{-e --stack-depth}	Specifies the number of methods to display in the command prompt. When unspecified, the command uses 10 as the default value.
{-s --server}	Executes the command on another server instance available on the local system. When unspecified, the <code>default</code> instance is used.
{-k --ssl}	Indicates if communication is over SSL. Valid arguments are:

Arguments and Option	Description
	<ul style="list-style-type: none"> ■ <code>true</code> use this value when communications are carried out over SSL. ■ <code>false</code> use this value when My webMethods Server is not configured to expose JMX over SSL (default).
<code>{-x --xml}</code>	Generates XML output. A linked XSL stylesheet is also generated to enable viewing the XML in a web browser (rendered as HTML).
<code>{-l --legacy}</code>	Generates text output in the format used in My webMethods Server version 9.6 and earlier. This format is not compatible with the third-party Thread Dump Analyzer tool.
<code>{-o --output}</code>	Saves the output to the specified file on the file system and does not display it in the console window.
<code>{-h --help}</code>	Displays command help text for the command, the available options and arguments, and default values.

Usage Notes

- By default, the `threaddump` command generates output in plain text format. Beginning with My webMethods Server version 9.7, the default text output can be consumed by the third-party Thread Dump Analyzer tool. Use the `-l | --legacy` option to generate text output in the previous format.
- The `threaddump` command uses JMX, which requires authentication. You must provide user credentials with the `threaddump` command to obtain the needed information. In addition, the provided user name must have the My webMethods Server Connect to MWS JMX functional privilege, `mws.privilege.jmx.connect`.
- The `threaddump` command saves the information in the `diagnostics.log` file located in the *Software AG_directory* `/MWS/tools/diagnostics/config/` directory.
- The `threaddump` command produces the result using the `log4j` mechanism. For more information, see [“Log4j” on page 31](#). The log file is located in the directory from which the tool was started. If you run the tool repeatedly, the logging utility appends the new information to the existing log file from the previous run.

Examples

- Windows example:

```
threaddump.bat -a all -n localhost -p 5002 -e 8 -s myServer -x -o
C:\softwareag\MWS\tools\diagnostics\bin\output\ -b user -g password
```

- UNIX example:

```
./threaddump -a all -n localhost -p 5002 -e 8 -s myServer -x -o
/opt/softwareag/MWS/tools/diagnostics/bin/output/ -b user -g password
```

dbintegritycheck

Description

The `dbintegritycheck` command reviews the dynamic business objects (DBOs) deployed to My webMethods Server to search for potential errors that might have occurred if a DBO was installed or upgraded incorrectly. You can run the `dbintegritycheck` command on Windows and UNIX at any time. In addition, My webMethods Server automatically executes `dbintegritycheck` at the following times:

- After a My webMethods Server fix installation.
- When a My webMethods Server fix is rolled back.
- After an over-install upgrade of My webMethods Server.
- During My webMethods Server startup.

To check the integrity of a deployed DBO, `dbintegritycheck` compares the VARCHAR column data type of the deployed DBO with the metadata for the DBO. If a difference is encountered, `dbintegritycheck` reports the discrepancy and provides the table name, column name, and data type associated with the discrepancy.

The `dbintegritycheck` command is located in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

Syntax

- On Windows: `dbintegritycheck.bat`
- On UNIX: `./dbintegritycheck`

Options

```
[{-l|--reportLevel} [quiet|full]]  
[{-o|--output} directory]  
[{-s|--server} name]  
[{-x|--xml}]  
[{-h|--help}]
```

The following table lists the arguments and options of the `dbintegritycheck` command-line tool:

Arguments and Options	Description
<code>{-l --reportLevel} [quiet full]</code>	Indicates the level of detail you want in the report. Specify one of the following: <ul style="list-style-type: none">■ <code>quiet</code> indicates that you want <code>dbintegritycheck</code> to list only the tables, columns, and data types

Arguments and Options	Description
	<p>that did not match, that is report only the discrepancies. This is the default.</p> <ul style="list-style-type: none"> ■ <code>full</code> indicates that you want <code>dbintegritycheck</code> to list all tables, columns, and data types and a message stating whether the information matched, indicating it is valid, or did not match, indicating a discrepancy.
<code>{-s --server} name</code>	<p>Executes <code>dbintegritycheck</code> on another My webMethods Server instance that is available on the local system. Supply the name of the server instance for <code>name</code> .</p> <p>By default, if you do not use the <code>{-s --server}</code> option to specify an alternative server instance, the tool executes on the default instance.</p>
<code>{-o --output} directory</code>	<p>Stores the resulting output to a directory you specify. Supply the directory path where you want the tool to write the output.</p> <p>By default, if you do not use the <code>{-o --output}</code> option, <code>dbintegritycheck</code> stores the results in the same directory in which you started the command.</p>
<code>{-x --xml}</code>	<p>Generates the results in XML files rather than displaying the results in the console window. The tool creates the following two XML files:</p> <ul style="list-style-type: none"> ■ Standard log4j XML report. You can execute <code>loganalyzer</code> against this XML file. ■ Structured XML report. You can use this XML file with the <code>envdiff</code> tool.
<code>{-h --help}</code>	<p>Displays information about the tool and the available options.</p>

Exit Codes

When `dbintegritycheck` exits, it provides one of the following exit code values:

Exit Code Value	Meaning
0	The <code>dbintegritycheck</code> tool found no errors in the database.
1	The <code>dbintegritycheck</code> tool found discrepancies in the database.

Exit Code Value	Meaning
2	The dbintegritycheck tool encountered errors unrelated to the database or cannot be executed for some reason.

Usage Notes

- When you execute dbintegritycheck, specify the `{-x | --xml}` option to have the results stored as XML files. You can then use other My webMethods Server commands to analyze the results. Specifically, you can use:
 - The loganalyzer command. For example, after you execute dbintegritycheck with the `{-x | --xml}` option so that it produces XML files, you can execute the loganalyzer command, specifying the location of the XML file that is in log4j format. For more information, see [“loganalyzer” on page 19](#).
 - The envdiff command. For example, you can compare the results of two different executions of dbintegritycheck with the envdiff command, as described in [“Usage Notes” on page 30](#). For more information, see [“envdiff” on page 17](#).
- My webMethods Server automatically executes dbintegritycheck at startup and logs the results to the `_full_.log` file. Although it is recommended that you keep this action enabled, you can prevent My webMethods Server from executing dbintegritycheck at startup. To do so, see the topic [“Preventing My webMethods Server from Executing dbintegritycheck at Startup”](#) in the PDF publication *Administering My webMethods Server*.

Examples

- To execute dbintegritycheck on Windows against the server instance named myServer and use the default reporting level that only shows discrepancies, specify the following command:

```
dbintegritycheck.bat -s myServer
```

- To execute dbintegritycheck on Windows against the server instance named myServer, set the reporting level to full, and have the tool write the results to XML files, specify the following command:

```
dbintegritycheck.bat -s myServer -l full -x
```

- To execute dbintegritycheck on UNIX against the default server instance and write the output to a directory that you specify, use the following command:

```
./dbintegritycheck -o /opt/softwareag/MWS/tools/diagnostics/bin/output
```

- To execute dbintegritycheck on UNIX against the default server instance, set the reporting level to full, and have the tool write the results to XML files, specify the following command:

```
./dbintegritycheck -l full -x
```

- After you install My webMethods Server, you can execute the following dbintegritycheck command to create a full report so that you have a baseline structured XML file for future reference. On Windows:

```
dbintegritycheck.bat -l full -x
```

Save a copy of the generated XML file by renaming the file to something like `baseline_db_integrity_check.xml`. If you encounter a problem in the future, execute the same `dbintegritycheck` command again to create a full report that contains the current database contents. You can then use the `envdiff` tool to compare the two reports to uncover any errors.

Log4j

My webMethods Server and its diagnostic tools use the [Apache log4j](#) open source API for logging, formatting, and adding log statements from the server and diagnostic commands. The `log4j` file and the tool are configured to save the results in log, XML, and zip files in the directory where you started the diagnostic tool.

The diagnostics tools use the `log4j.xml` configuration file to save and format results in diagnostics and error logs.

The `log4j.xml` file is located in the following directory:

```
Software AG_directory\MWS\tools\diagnostics\config\log4j.xml.
```

The following command-line tools use the `log4j.xml` file to configure the output.

- The `threaddump` tool reports results in the `diagnostics.log` file.
- The `envcapture` tool reports results in the `diagnostic.log` file.

The following command line tools use the `log4j` mechanism:

- [“envcapture” on page 14](#)
- [“logalyzer” on page 19](#)
- [“threaddump” on page 25](#)

Java Artifact Mapping in My webMethods Server

At run time, My webMethods Server creates a relationship between logged issues and the mapped components and products that are deployed on the server.

Whenever a server instance is created, a new layered product is installed, or during an upgrade of My webMethods Server, the Software AG Installer automatically invokes the following command, which generates a new My webMethods Server class path:

```
/MWS/bin/mws.bat -s server_instance_name update
```

When the server is updated, a `javaArtifactMapping.xml` file is also created and stored in the installation directory at:

```
Software AG_directory/MWS/server/server_instance_name/config
```

The `javaArtifactMapping.xml` file contains the mapping between all of the available Java classes or packages and their corresponding Java artifacts such as jars, portlets, and web applications.

The loganalyzer command-line tool uses the updated `javaArtifactMapping.xml` file when it analyzes and displays the information. For more information, see [“loganalyzer” on page 19](#).

Important:

If any custom components are deployed into the My webMethods Server instance after you executed the update command, the `javaArtifactMapping.xml` might not contain the information about the new custom components. For more information, see [“Updating the Java Artifact Mapping File” on page 32](#).

The `javaArtifactMapping.xml` file stores the mapping information about the installed JAR files and layered products in the following elements:

```
<jar name="JAR_name" relative_path="install_path" build_number="value"
  productId="product_install_ID">
<class name="class_name"/>
.
.
. </jar>
```

```
<product id="product_install_ID" displayName="product_display_name"
  version="product_version"/>
```

Updating the Java Artifact Mapping File

If you deploy custom components into the My webMethods Server without executing the update command, the `javaArtifactMapping.xml` file will not contain information about the newly deployed custom components. As a result, mapping information will not be available in the logging results for those components.

➤ To update the `javaArtifactMapping.xml` file

1. Stop My webMethods Server.
2. Open a command prompt in `Software AG_directory/MWS/bin`.
3. Enter this command:

```
mws -s server_instance_name update
```

4. Start My webMethods Server.

3 Using External Command-Line Diagnostic Tools

- Overview of External Command-Line Diagnostic Tools 34
- Using jconsole to Diagnose My webMethods Server 34
- Using jconsole in Standard Mode 35
- Using jvisualvm to Diagnose My webMethods Server 35

Overview of External Command-Line Diagnostic Tools

My webMethods Server diagnostic command-line tools include interfaces for external applications that are distributed with the installation of the Oracle JDK:

- `jconsole`
- `jvisualvm`

You can use these command-line tools to diagnose My webMethods Server. You can also use these external tools in their standard mode. For more information about `jconsole` and `jvisualvm`, see:

- [“Using `jconsole` to Diagnose My webMethods Server ” on page 34](#)
- [“Using `jconsole` in Standard Mode” on page 35](#)
- [“Using `jvisualvm` to Diagnose My webMethods Server ” on page 35](#)

Using `jconsole` to Diagnose My webMethods Server

The `jconsole` command-line tool is installed on the system as part of the JDK. The My webMethods Server installation contains an executable batch file (`jconsole.bat`) and a shell script (`jconsole`) in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

The following procedure uses these executable files to start `jconsole` in diagnostic mode for My webMethods Server. To run `jconsole` in its standard mode, see [“Using `jconsole` in Standard Mode” on page 35](#).

➤ To start the `jconsole` tool to diagnose My webMethods Server

1. Open a command console and `cd` to this directory:

Software AG_directory /MWS/tools/diagnostics/bin

2. Start the executable with this command:

- Widows:

```
jconsole
```

- UNIX:

```
./jconsole
```

As installed, the command executes and applies these parameters for SSL authentication:

```
-J-Djavax.net.ssl.trustStore=%SERVER_HOME%\config\security  
\sagdemoca.jks -J-Djavax.net.ssl.trustStoreType=jks
```

This enables `jconsole` to connect to My webMethods Server.

Using jconsole in Standard Mode

The jconsole command-line tool is installed on the system as part of the JDK. The My webMethods Server installation contains an executable batch file (jconsole.bat) and a shell script (jconsole) in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

The following procedure uses these executable files to start jconsole in standard mode which means that it is not connected to My webMethods Server. This is useful when you want to use jconsole with Java applications other than My webMethods Server. To run jconsole in its My webMethods Server diagnostic mode, see [“Using jconsole to Diagnose My webMethods Server” on page 34](#).

➤ To start the jconsole tool in standard mode

1. Open a command console and cd to this directory:

Software AG_directory /MWS/tools/diagnostics/bin

2. Start the executable with this command:

- Windows:

```
jconsole -s false
```

- UNIX:

```
./jconsole -s false
```

The `-s false` option disables the security settings for My webMethods Server and prevents jconsole from connecting to My webMethods Server. This is equivalent to running jconsole from the Java distribution.

Using jvisualvm to Diagnose My webMethods Server

The My webMethods Server diagnostic tools installation contains an executable batch file (jvisualvm.bat) and a shell script (jvisualvm) in the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

The jvisualvm command-line diagnostic tool for My webMethods Server starts the VisualVM tool in My webMethods Server diagnostic mode. The jvisualvm diagnostic tool requires a separate installation of the VisualVM tool, or a JDK that includes the tool.

The following procedure uses these executable files to start VisualVM in diagnostic mode for My webMethods Server.

➤ To start the jvisualvm tool in My webMethods Server diagnostic mode

1. Open a command console and navigate to the following directory:

Software AG_directory /MWS/tools/diagnostics/bin

2. To start the executable, run the following command:

- Windows:

```
jvisualvm -p visualvm_home
```

- UNIX:

```
./jvisualvm -p visualvm_home
```

where *visualvm_home* is the fully qualified path to the installation directory of the VisualVM tool.

As installed, the command executes with these parameters for SSL authentication:

```
-J-Djavax.net.ssl.trustStore=%SERVER_HOME%\config\security  
\sagdemoca.jks -J-Djavax.net.ssl.trustStoreType=jks
```

This enables VisualVM to connect to My webMethods Server.

4 Using My webMethods Server Diagnostic Portlets

■ Using the Thread Dump Portlet	38
■ Using the Performance Analysis Portlet	39
■ Using the Performance Statistics Portlet	43
■ Using the Log Analysis Tools	46
■ Using the Memory Monitor Portlet	52

My webMethods Server Monitoring and Diagnostic portlets are available on the **Tools** tab on the administrator's panel of My webMethods Server. For more information about using My webMethods Server, see *Working with My webMethods*.

Using the Thread Dump Portlet

You use this portlet to monitor thread execution deadlocks in My webMethods Server threads. The tool uses the same Java virtual machine (JVM) as My webMethods Server and obtains the thread dump information. The tool relies on a JVM algorithm to detect deadlocks on My webMethods Server.

However, as the JVM does not always reliably detect deadlocks, additional thread dump analysis may be necessary. To do that, the tool displays not only deadlock threads but also all available threads. If the tool does not display any deadlock threads, it is recommended that you review the complete thread dump information and use it for further analysis.

If the tool detects a deadlocked thread, it is recommended that you export the thread dump information to a file and forward it to the appropriate support group for the organization that did the implementation.

Displaying Thread Information

> To display My webMethods Server thread information

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.
3. Locate the **Thread Dump** tool and drag it onto the workspace.
4. To display the most recent thread information, click **Refresh**.

Filtering Thread Information

> To narrow the thread information of My webMethods Server in the result

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.
3. Locate the **Thread Dump** tool and drag it onto the workspace.
4. Enter filtering criteria in the **Filter by Thread Name** field.
5. Choose **Go**.

Exporting Thread Information

➤ To export My webMethods Server thread information

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.
3. Locate the **Thread Dump** tool and drag it onto the workspace.
4. Select the **Export to File** button.
5. Specify a directory on the file system in which the tool saves the result into an XML file.

Using the Performance Analysis Portlet

The **Performance Analysis** tool enables portlet developers or administrators to measure and analyze the performance of My webMethods Server services and custom portlets or applications using an embedded performance-monitoring service. The tool enables you to:

- Manage the available performance-monitoring services. Depending on the service that you configure and use, the tool displays lightweight or complete performance information.
- Capture a particular snapshot and then analyze the performance within the particular interval.
- Export the snapshot into an XML file. At a later stage, you can provide the exported data for further analysis, import it and analyze it on another system or save it as reference information.
- Export the analyzed data into a comma separated values (CSV) file. However, you cannot import or analyze CSV files with the tool.

As a developer, you use the tool to simultaneously monitor and analyze the performance of the applications which you create. The performance service that you use provides thorough analysis and enables you to browse the details of the particular implementation. As an administrator, you use it to analyze the overall performance of My webMethods Server and facilitate for further assistance or troubleshooting.

Configuring Performance Service

➤ To configure performance service

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.

3. Locate the **Performance Analysis** tool and drag it onto the workspace.
4. On the tool menu, click **Properties**.
5. Click the **Preferences** tab if it is not already selected. In the **Performance Service** area, click **Stop** to stop the performance analysis service.
6. Configure the performance service that you want to use with My webMethods Server. The following options are available:

- **JMX Monitoring Service - Suitable for use in production**

This service starts a lightweight monitoring mode. In this mode, the service provides limited features, yielding improved runtime performance. The JMX monitoring service does not allow expanding or selecting a token and does not provide information about parent or child token relations in the results. This prevents selection of concrete parent entries from the **Refine** drop-down menu. The tool does not display parent or child related information for the tokens in the results table.

This is the default service selection. The service constantly monitors performance characteristics of My webMethods Server in production or in any other environment.

The JMX Monitoring Service also makes performance information available with JMX. If My webMethods Server is running in monitoring mode, you can use any JMX client, for example the default JDK 1.6 jVisual VM tool, to connect to My webMethods Server on port 5002 (by default) to monitor performance service data.

- **Development Performance Service - Significant performance penalty for production use!**

This option starts a more thorough analyzing service which results in slower performance. The development performance service allows the tool to display parent and child token relations and thus enable better performance analysis. If you start this service, it enables you to choose and browse for concrete parent entries from the **Refine** drop-down menu. The tool displays the parent tokens in the results table and enables you to browse performance information for the child tokens.

- **Make Service Type and State configuration persistent (effective after server restart)**

Click this check box to make the performance service mode persistent. That is, this mode is in effect each time that My webMethods Server starts. Otherwise, the selected configuration remains effective until you change it manually or until you restart My webMethods Server. If you do not select this option, the configuration is reset to default upon the next restart.

7. Click **Apply** to save your changes.
8. Click **Start** to start the performance analysis service.

Configuring the Displayed Content

➤ To configure the content that the tool displays

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Performance Analysis** tool and drag it onto the workspace.
4. On the tool menu, click **Properties**.
5. On the **Display Defaults** area of the **Preferences** window, configure the display options of the tool. You can configure the following options:

These settings provide the initial configuration of the tool and are used any time when the tool is started and displayed. You can change some of the display options dynamically on the tool user interface, although in this case the settings are not persistent and remain valid only for the current session of the tool.

■ **Number of Rows to Display**

From the drop down menu, choose the number of rows which the result table contains.

■ **Sort By**

From the drop down menu, choose the sort criteria. You can sort the result by average time, name, number of actions, or total time. You can sort the table any time by clicking on the column header.

Note:

If you start the development performance service, the following sort options are also available: parent schema compare actions, parent schema compare time, and parent time.

■ **Sort Order**

The available options enable you to sort the result ascending or descending. You can change the sort order any time by clicking on the table column header.

6. On the **Column Display** area, configure the columns which you want to display on the result table. Moreover, you can configure the order of the columns. You can change the displayed columns or their order any time by dragging a column and dropping it into a new location. If you do not specify any columns explicitly, the tool displays all by default.
7. Save your changes. The changes are persisted only for the current instance of the tool on this workspace.

Displaying Performance Analysis Information

➤ To display performance analysis information

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Performance Analysis** tool and drag it onto the workspace.
4. From the **Refine** drop-down menu, specify the content category which you want to display.
5. To display the most recent analysis information, click **Refresh**.
6. To reset the content that the tool displays and continue capturing statistic information, click **Reset**.
7. To sort out the content that the tool displays, click the respective table header. The tool enables you to sort the content ascending or descending. When the **Development Performance Service** is enabled, you can navigate in and out between parent or child tokens using the token and breadcrumb entries.

Exporting Displayed Content

➤ To export the content that the tool displays

1. On the **Performance Analysis** tool, click **Export**.
2. On the dialog that opens, specify whether you want to save the exported file on the file system, or only open it.

The tool exports a snapshot in XML format. The snapshot contains the complete information that is available for all categories.

Importing and Analyzing Snapshot Information

➤ To import and analyze snapshot information using the tool

1. On the **Performance Analysis** tool, click **Import**.
2. On the dialog that opens, specify an XML file which you want to analyze and then click **Import**.

The tool imports the XML file and if the file structure contains performance information with the expected format, the tool analyzes and displays the snapshot information.

Exporting the Table Content in Comma Separated Values Format

➤ To export the table content that the tool displays in a Comma Separated Values file

1. On the **Performance Analysis** tool, click **Export Table**.
2. On the dialog that opens, specify the encoding for the exported data and then click **Export**.
3. Specify a directory on the file system in which you want to download and save the file and then click **OK**. Alternatively, you can open the file without saving it on the file system.

The tool exports a snapshot in CSV format. The snapshot contains the complete information that is available in the result table for a corresponding category or a parent token.

Using the Performance Statistics Portlet

My webMethods Server provides a set of preconfigured tools that display statistic and analysis information about the performance of various My webMethods Server actions which are grouped by categories. As an administrator, you use the **Performance Statistics** tool to monitor category entries, such as portlets, Web service client calls, database calls, and so on. Depending on the settings, which you configure, the tool displays the slowest or the fastest performing entries, the most invoked entries, or the average time which the respective entry needs to perform an action.

Configuring the Displayed Content

➤ To configure the content that the tool displays

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Performance Statistics** tool and drag it onto the workspace.
4. On the tool menu, click **Properties**. The **Preferences** window opens.
5. On the **Output Settings** area, configure the following options:
 - **Category**

From the drop-down menu, chose the category for which you want to display performance statistic information.

■ **Output Type**

From the drop down menu, choose how you want to display the information. Depending on your choice, you can change the tool interface to a table view, a pie chart, or a bar chart.

Note:

The tool enables you to export the statistic information to a file, only when displayed in a table view.

■ **Number of Items**

From the drop down menu, choose the number of items for which you want to display statistic information.

6. On the **Display Defaults** area, configure the display options of the tool. You can configure the following options:

■ **Sort Order**

The available options enable you to sort the result ascending or descending.

■ **Sort By**

From the drop down menu, choose the sort criteria. You can sort the result by average time, name, number of actions, or total time.

7. On the **Column Display** area, configure the columns which you want to display on the tool. If you do not specify any columns explicitly, the tool displays all by default.

8. Save your changes.

Displaying Performance Statistics Information

➤ To display performance statistics information

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Performance Statistics** tool and drag it onto the workspace.
4. To display the most recent statistics information, click **Refresh**.

Note:

The tool supports both JMX and developer performance service implementations.

Exporting the Content in Comma Separated Values Format

➤ To export the content that the tool displays in a Comma Separated Values file

1. Start the **Performance Statistics** tool in a table view mode.
2. Click **Export Table**.
3. On the dialog that opens, specify the encoding for the exported data and then click **Export**.
4. Specify a directory on the file system in which you want to download and save the file and then click **OK**. Alternatively, you can open the file without saving it on the file system.

To export a complete performance snapshot in XML format use the Performance Analysis tool. For more information, see [“Using the Performance Analysis Portlet” on page 39](#) .

Displaying Slow Database Calls

➤ To display information about ten of the slowest database calls

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Slow Database Calls** tool and drag it onto the workspace.

Displaying Slow Performing Portlets

➤ To display performance information about ten of the slowest portlets

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Slow Portlets** tool and drag it onto the workspace.

Displaying Slow Performing Web Services

➤ To display performance information about ten of the slowest Web services

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Performance Information**.
3. Locate the **Slow Web Services** tool and drag it onto the workspace.

Using the Log Analysis Tools

My webMethods Server provides a set of preconfigured log analysis tools that read and analyze log files created within My webMethods Server in accordance with the log4j mechanism. As an administrator, you configure centrally a common data source file and analysis time period which is then used in all log analysis tools.

Configuring Log Analysis Data Sources

> To configure the log analysis data source

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Log Tracing and Analysis**.
3. Locate the **Log Analysis Datasource** tool and drag it onto the workspace.
4. On the **Add Logs** option list, proceed as follows:

Note:

To use and analyze the default errors.log file of your My webMethods Server, choose the corresponding entry on the **Select Log File** list and do not upload additional log files.

■ Upload Local File

To use and analyze an external log file that is available on the local file system, choose this option and then upload a file.

■ Download from External URL

To use and analyze an external log file that is available on a URL address, choose this option and then provide the URL of the file.

If you upload a proper log file, the tool adds it to the **Select Log File** list so you can select and analyze it. The tool analyzes the errors.log file of My webMethods Server which is an XML file created in accordance with the log4j mechanism.

If several log files are contained in a zip archive, you can upload the archive. The tool examines the log files in the zip file and displays them. The tool does not examine plain text log files.

You can use this feature with archives created by the `envcapture` command line tool. For more information, see [“envcapture” on page 14](#).

5. On the **Select Log File** list, choose the log file which you want to analyze.

You can switch between several log data sources by selecting one of them and clicking **Apply**. All log analysis portlets which you opened on the workspace refresh their content and display the information from the selected data source. You can use this technique to compare two or more snapshots of My webMethods Server logs.

6. Click **Apply**.

Configuring the Log Analysis Time Period

> To configure the log analysis time period

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Log Tracing and Analysis**.
3. Locate the **Log Analysis Time Period** tool and drag it onto the workspace.

4. On the **Date Range** drop-down menu, choose the period range which you want to set.

The default time period is set to include the entire date range captured by the selected logs datasource.

5. Optionally, on the **Start Date** and **End Date** fields, select a custom time period which you want to set.

If you select a time period that goes beyond the captured date range, the logs data is displayed only for what is captured in the logs.

6. Click **Apply**.

Viewing Log Messages with the Log Viewer Tool

The Log Viewer tool enables you to view the latest messages in the plain text log files of My webMethods Server. You can set the log messages collection criteria using the Log Configuration tool. For more information, see the *Analysis, Reporting and Troubleshooting* chapter of *Administering My webMethods Server*.

> To view log messages

1. Log on as an administrator on My webMethods Server.

2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Log Tracing and Analysis**.
3. Locate the **Log Viewer** tool and drag it onto the workspace.
4. In the **Log File** field, select the file that you want to analyze.
5. In the **Lines** field, select the number of lines that you want to display.

The default configuration of the Log Viewer tool is set to show the last page of the last 500 lines of My webMethods Server full log file.

The list of files is updated whenever the logs are rotated. This happens either overnight or based on the log file size depending on the log4j configuration. You can use the updated list of files when you re-open the tool. For more information, see Logging Configuration

6. To display the most recent information, click **Refresh**.

You can sort the table by any column. If a log entry has a stack trace associated with it, you can display it by clicking on the icon in the **ST** column.

7. To export the provided plain text log files, click **Export Logs** and proceed as follows:
 - a. From the **Available** area, on the window that opens, choose a file which you want to export.

If you select a single log file for export, the application enables you to choose whether you want to compress it into a zip file. If you do not select the **Compress selected log files before export** option, the downloaded data is an uncompressed plain text file.

When you select and export more than one log file, the application automatically compresses the data into a zip file.

- b. Click **Export**.
 - c. Specify a directory on the file system in which you want to download and save the file and then click **OK**. Alternatively, you can open the file without saving it on the file system.
8. To search for a particular entry in the result table, in the **Text** and **Date** fields provide your search criteria and proceed as follows:

- a. To display the first entry that matches the search criteria, click **Search**.
 - b. To display a consecutive entry that matches the search criteria, click **Search Next**.

The tool highlights the found entry.

- c. To reset the search criteria, click **Clear**.

Using the Predefined Logging Tools

My webMethods Server provides a set of pre-configured tools that enable you to perform basic log analysis. The tools provide information about the products and components that caused the log issues.

To map the logging information to the corresponding products, My webMethods Server uses the information that is stored in the `javaArtifactMapping.xml` file. For more information about the Java mapping mechanism, see [“Java Artifact Mapping in My webMethods Server ”](#) on page 31.

➤ To start the predefined My webMethods Server logging tools

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > Log Tracing and Analysis**.
3. Select a tool to work with and drag it onto the workspace:

The following table lists the diagnostic tools, available for selection:

Tool	Description
Log Errors by Severity	<p>The tool displays statistics about the logs in the configured data source file which occur in the configured time period. Information about the total appearances of the issues is displayed, as well as the time interval in which the issues appeared. The tool analyzes and displays only logs of the following severity:</p> <ul style="list-style-type: none"> ■ WARNING ■ ERROR ■ FATAL <p>To display thorough information about a particular severity type, choose the respective graphic that is displayed on the tool. Depending on your selection, the tool opens the Log Issues tool and displays information about the logs of the respective severity.</p> <p>The Log Issues tool that opens after you choose a graphic bar enable you to display more thorough issue details. You can choose a log message from the table and then on the Issue Details window that opens, you can click and browse for more information the recorded timestamps.</p>
Log FATAL Issues	<p>The tool displays statistics about the logs in the configured data source file which occur in the configured time period. The tool analyzes and displays only logs of FATAL severity. Information</p>

Tool	Description
	<p>about the total appearances of the issue is displayed, as well as the time interval in which the issue appeared.</p> <p>To display thorough information about a particular log, choose the respective log message that is displayed on the tool. The tool opens the Issue Details window (see Issue Details description below).</p>
Log Issues	<p>The tool displays information about the logs in the configured data source file which occur in the configured time period. Information about the total appearances of the issue is displayed, as well as the time interval in which the issue appeared. The tool analyzes and displays only logs with the following severity:</p> <ul style="list-style-type: none"> ■ WARNING ■ ERROR ■ FATAL <p>To display thorough information about a particular log, choose the respective log message that is displayed on the tool. The tool opens the Issue Details window (see Issue Details description below).</p>
Log Issues by Component	<p>The tool displays statistics about the logs in the configured data source file which occur in the configured time period. The tool analyzes and displays the logs by the component in which they appear. Information about the component of the issue is displayed, as well as the time interval in which the issue appeared.</p> <p>To display thorough information about a particular component type, choose the respective graphic bar that is displayed on the tool. Depending on your selection, the tool opens the Issues For Component window and displays information about the logs of the respective component.</p> <p>The Issues For Component tool that opens after you choose a graphic bar enable you to display more thorough issue details. You can choose a log message from the table and then on the Issue Details window that opens, you can click and browse for more information the recorded timestamps.</p>
Log Issues by Product	<p>The tool displays statistics about the logs in the configured data source file which occur in the configured time period. The tool analyzes and displays the logs by the product in which they appear. Information about the product of the issue is displayed, as well as the time interval in which the issue appeared.</p>

Tool	Description
	<p>To display thorough information about a particular product, choose the respective graphic that is displayed on the tool. Depending on your selection, the tool opens the Issues For Product tool and displays information about the logs of the respective product.</p> <p>The Issues For Product tool that opens after you choose a graphic bar enable you to display more thorough issue details. You can choose a log message from the table and then on the Issue Details window that opens, you can click and browse for more information the recorded timestamps.</p>
Issue Details	<p>The Issue Details window opens when you click a log issue on the following tools:</p> <ul style="list-style-type: none"> ■ Log FATAL Issues ■ Log Issues <p>The window displays the following information:</p> <ul style="list-style-type: none"> ■ Message.Text of the log message. ■ Count. Number of occurrences and the similarity level given in percent. The percent shows the similarity of the least equivalent log message included in and compared to the currently selected issue. ■ Component. A list of components in which the log issue appears. For more information about the mapping between the log message and the corresponding stack trace, see “Java Artifact Mapping in My webMethods Server ” on page 31. ■ Product. A list of products in which the log issue appears. For more information about the mapping between the log message and the corresponding stack trace, see “Java Artifact Mapping in My webMethods Server ” on page 31. ■ Count by day. A graph that displays the count of occurrences by day. ■ Timestamp. A list of occurrences for a particular date in the graph. To change or refresh the list of timestamps, click on the corresponding date bar in the graph. You can click a specific timestamp to open the Log Viewer with the selected entry highlighted in the results table. For more information about working with the log viewer, see “Viewing Log Messages with the Log Viewer Tool” on page 47.

Tool	Description
Most Used Portlets	The tool displays statistics about ten of the most used portlets in My webMethods Server. To display the most recent information, click Refresh .
Most Used Web Services	The tool displays statistics about ten of the most used Web services in My webMethods Server. To display the most recent information, click Refresh .

- For more information about configuring data source log files, see [“Configuring Log Analysis Data Sources”](#) on page 46.
- For more information about configuring logging time period for analysis, see [“Configuring the Log Analysis Time Period”](#) on page 47.
- For more information about performance statistics of My webMethods Server, see [“Displaying Performance Statistics Information”](#) on page 44.

Exporting Displayed Content for Log FATAL Issues and Log Issues

The **Log FATAL Issues** and **Log Issues** tools enable you to export the displayed information to a file:

➤ To export the table content in comma separated value format

1. In the Log FATAL Issues or Log Issues windows, click **Export Table**.
2. In the resulting dialog box, specify the encoding for the exported data and then click **Export**.

A browser-generated dialog box appears that enables you to either specify a directory where you can save the file, or open the file without saving it on the file system.

3. Click the save or open option, and then click **OK**.

Using the Memory Monitor Portlet

As an administrator, you use the Memory Monitor tool to monitor the memory usage of the Java virtual machine (JVM) and send e-mail notifications to administrators when the configured memory threshold limits are reached.

Note:

No e-mail notifications are sent until at least one e-mail address is configured.

The following thresholds must be set:

- **WARNING.** The amount of free JVM memory that triggers a warning notification.

- **ERROR.** The amount of free JVM memory that triggers an error notification.
- **FATAL.** The amount of free JVM memory that triggers a fatal notification and, if so configured, a server restart.

For the **WARNING** and **ERROR** thresholds, the portlet sends a notification to the configured e-mail accounts each time the specified memory usage is met.

When the **FATAL** threshold is met, the portlet also sends an e-mail notification. However, you can optionally specify if the **FATAL** threshold should also restart the server. In this case, heap dumps and thread dumps are created just prior to the restart.

An additional notification is sent after a successful restart of My webMethods Server. The notification indicates memory usage statistics before and after the restart.

Note:

In some cases, Memory Monitor reporting might not function as expected. For example, it might not prevent an `OutOfMemory` exception by never reaching the **FATAL** threshold, due to dependencies on the threshold configuration, Java Runtime version, and the specific configuration of the current production environment. Therefore, to achieve desired results, you may need to experiment with the interaction of Memory Monitor configuration and the Java dependencies noted in the previous paragraph.

Configuring the Memory Monitor Service

➤ To configure the content that the tool displays

1. Log on as a system administrator on My webMethods Server.
2. In My webMethods Server: **Folders > Administrative Folders > Administration Dashboard > Configuration > CAF Application Runtime Configuration.**
3. On the **Keyword** tab, type `wm_mws_diagnostics` in the **Keywords** field.
4. Click **Search**.
5. In the results list, click the **wm_mws_diagnostics** entry.
6. Click **Web Application > Environment Entries**
7. On the **Web Application - Environment Entries** area that opens on the right hand side, configure the following settings:

Important:

When you configure the following thresholds, make sure that the **WARNING** threshold has the greatest memory amount, followed by the **ERROR** threshold, and finally the **FATAL** threshold that has the smallest memory amount.

■ thresholdWarn

Specify the amount of free memory of the JVM (in MB) that triggers a warning message notification to the configured e-mail accounts. The default is 150 MB.

■ thresholdError

Specify the amount of free memory of the JVM (in MB) that triggers an error message notification to the configured e-mail accounts. The default is 100 MB.

■ thresholdFatal

Specify the amount of free memory of the JVM (in MB) that triggers a fatal message notification to the configured e-mail accounts. The default is 50 MB.

■ restartServer

In the text field, type `true` to enable server restart when the **thresholdFatal** setting is reached. Set to `false` by default. In addition, heap dumps and thread dumps are generated in *Software AG_directory \MWS\server\ServerInstance\logs\snapshots* for `threaddump-date_EDT.hprof`.

■ enabled

Accept the default value of `true` to enable the Memory Monitor tool, or set this field to `false` to disable the tool. This setting is not affected by the **Start** and **Stop** buttons on the portlet's **Properties** tab in My webMethods, which control only the display of memory information.

■ email

Type the e-mail accounts that will receive e-mail notifications. Use a comma (,) to separate multiple addresses.

■ thresholdPercent

Set this field to `true` to interpret the **thresholdWarn**, **thresholdError**, and **thresholdFatal** setting values as percentages; set this field to `false` to interpret the threshold setting values as megabytes. When set to `true`, the **thresholdWarn**, **thresholdError**, and **thresholdFatal** values must be between 1 and 100. The default setting is `false`.

8. Click **Apply**.

Displaying Memory Monitor Statistics Information

> To display memory statistics and configuration information

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.

3. Locate the **Memory Monitor** tool and drag it onto the workspace.

The tool displays the following information:

- **Max Memory**

This indicates the maximum usable memory amount from the JVM memory.

- **Allocated Memory**

This indicates the total memory amount reserved by the JVM.

- **Free Memory**

This indicates the free memory amount that is available on the JVM.

- **Used Memory**

This indicates the used memory amount on the JVM.

- **Memory Usage graphic**

The graphic displays the average memory usage on the JVM in the past 24 hours.

Additionally, the tool displays information about the memory allocation in percents.

Starting and Stopping the Memory Monitor Service Display in My webMethods

The **Start** and **Stop** buttons do not affect the portlet's **enabled** setting in My webMethods Server. They control only the display of memory information.

» To start or stop the memory monitor service

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.
3. Locate the **Memory Monitor** tool and drag it onto the workspace.
4. On the tool menu, click **Properties**.
5. Click the **Preferences** tab if it is not already selected. The **Memory Monitor Service Timer** area displays the current configuration of the Memory Monitor tool.
6. Click the **Start** or **Stop** button to start or stop the memory monitor service.

Viewing the Memory Monitor Configuration Settings in My webMethods

You can set and view configuration options as a system administrator in My webMethods Server, as described in [“Configuring the Memory Monitor Service” on page 53](#). You can also view the configuration settings while working with the Memory Monitor portlet in My webMethods.

➤ To view Memory Monitor configuration settings in My webMethods

1. Log on as an administrator on My webMethods Server.
2. In My webMethods Server: **Tools > MWS Monitoring and Diagnostics > System Information**.
3. Locate the **Memory Monitor** tool and drag it onto the workspace.
4. On the tool menu, click **Properties**.
5. Click the **Preferences** tab if it is not already selected. The **Memory Monitor Service Timer** area displays the current configuration of the Memory Monitor tool.

The information in this area is read-only, displaying the notification threshold levels, e-mail notification addresses, and whether server restart is enabled or disabled. For more information about setting these configuration options, see [“Configuring the Memory Monitor Service” on page 53](#).

5 Troubleshooting My webMethods Server Using Diagnostic Tools

■ Troubleshooting Configuration Issues	58
■ Troubleshooting Environment Issues	64
■ Troubleshooting Performance Issues	66
■ Troubleshooting Log Issues	68

Troubleshooting Configuration Issues

My webMethods Server has a set of tools that enable you to easily identify and troubleshoot possible issues that you may encounter while working with it. This section presents some common configuration problems that are addressed by the diagnostic tools. The tools facilitate collecting the needed troubleshooting data and additionally provide output files that can be used by a support organization, such as Software AG Global Support or your internal support group.

The following table outlines potential problems, a description of the problem, and a reference to the tool that assists in the resolution:

Problem	Description	Reference
You cannot diagnose the threads of My webMethods Server.	To diagnose My webMethods Server, you must start it in monitoring mode.	For more information, see “Command Line Diagnostic Tools Requirements” on page 10.
The diagnostics.log file does not contain the thread dump information collected by the threaddump command line tool.	Depending on the configuration of the log4j.xml file, the tool stores the result into a log file or does not store it at all.	Configure the log4j.xml file.
You need external support for your My webMethods Server threads.	When you diagnose My webMethods Server threads, you can create an XML output file. You can use the <code>-o</code> (<code>--output</code>) and <code>-x</code> (<code>--xml</code>) arguments to export the diagnostic information.	For more information, see “threaddump” on page 25.
The following exception occurs when you invoke functionality remotely (using RMI or JMX): Connection refused to host: 127.0.0.1; nested exception is: java.net.ConnectException: Connection refused: connect	You use the <code>RMI_HOSTNAME</code> property when you invoke functionality (start threaddump or memorydump tools) using RMI or JMX on remote UNIX servers. If not set, the remote system may refuse the connection, or may not be recognized by the tools.	Set a correct host name or IP address to accept remote connections and invoke functionality remotely (using RMI or JMX). For more information, see the Modifying Configuration Files section of the PDF publication <i>Administering My webMethods Server</i> .
The following exception occurs when you invoke functionality remotely (using RMI or JMX): java.rmi.NoSuchObjectException: no such object in table	You use the <code>RMI_HOSTNAME</code> property when you invoke functionality (start threaddump or memorydump tools) using RMI or JMX on remote UNIX servers. If not set, the remote	Set a correct host name or IP address to accept remote connections and invoke functionality remotely (using RMI or JMX).

Problem	Description	Reference
	system may refuse the connection, or may not be recognized by the tools.	For more information, see the Modifying Configuration Files section of the PDF publication <i>Administering My webMethods Server</i> .
The displayed server threads report is too big and you cannot find a certain entry.	The server threads report can contain a large number of entries. To narrow the displayed information, you can filter on the thread names for specific entries. The filtering input is not case-sensitive.	For more information, see “Using the Thread Dump Portlet” on page 38 .
You need a file that contains the server threads report to request further assistance.	You can export thread information to a file on the file system and use it for further assistance by external support organization.	For more information, see “Using the Thread Dump Portlet” on page 38 .
You need to collect information about the My webMethods Server environment.	When you request assistance from the Software AG Global Support, you are asked to provide environment-specific information for My webMethods Server. To capture this information, you use the envcapture executable tool that is distributed with the installation of your server.	For more information, see “envcapture” on page 14 .
You need a snapshot of your file system structure after a successful install or update/upgrade.	You can capture environment information and then use it as a checkpoint after a successful installation, update, upgrade, configuration, or deployment of My webMethods Server.	For more information, see “envcapture” on page 14 .
You experience network connectivity problems.	You can capture information about the Internet connection settings.	For more information, see “envcapture” on page 14 .
You experience database connectivity problems.	You can capture information about the database connection.	For more information, see “envcapture” on page 14 .
You need to report information about the following:	The envcapture command line tool captures the required information.	For more information, see “envcapture” on page 14 .
<ul style="list-style-type: none"> ■ Log files. 		

Problem	Description	Reference
<ul style="list-style-type: none"> ■ Configuration files. ■ Installation folder structure, size, time stamps. ■ File system permissions for the current operating system user account. ■ Details about the local operating system and Java virtual machine. ■ Details about the Internet connectivity and the connectivity to database. 	<p>The tool uses the standard <code>System.getProperties()</code> and <code>System.getenv()</code> methods to obtain this information and include it in a report.</p>	
<p>The report file created by the <code>envcapture</code> tool does not contain a specific report.</p>	<p>Configure the <code>-a --action</code> option for the information type(s) that you want to capture. The default value is <code>all</code>.</p>	<p>For more information, see “envcapture” on page 14.</p>
<p>You want to compare snapshots created by the <code>envcapture</code> command line tool.</p>	<p>You can compare XML files that are part of the result of the <code>envcapture</code> tool.</p>	<p>For more information, see “envcapture” on page 14.</p>
<p>You encounter issues related to environment changes.</p>	<p>You can identify and troubleshoot issues related to environment changes.</p>	<p>For more information, see “envcapture” on page 14.</p>
<p>You need to find out what files have been added, updated, or deleted under the installation directory</p>	<p>You can use the <code>envdiff</code> tool to examine the structure of the XML files and outline the differences in the corresponding attributes:</p> <ul style="list-style-type: none"> ■ Folder structure ■ File size ■ MD5 hash ■ Timestamp ■ Permissions 	<p>For more information, see “envdiff” on page 17.</p>
<p>The reported differences within the XML files generated by the <code>envcapture</code> tool are too</p>	<p>You can manually exclude artifacts from the comparison criteria. Using the options and arguments that are available to</p>	<p>For more information, see “envcapture” on page 14.</p>

Problem	Description	Reference
comprehensive or too superficial.	the tool, you can exclude specific directory sub trees by specifying the root directory name of the sub tree, or attributes from the comparison and therefore from the result.	
You encounter problems when you start a tool in a directory on the file system that differs from the default one.	You use the default location to start the diagnostic tools. To invoke a tool from an alternative location on the file system, make sure that you set the WM_HOME global environment variable to point to <i>Software AG_directory</i> .	Configure environment variables.
You need to diagnose access permissions of directories.	The envdiff tool compares the user permissions of the directories and displays the difference in the result.	For more information, see “envdiff” on page 17.
Request further assistance from an external support organization about the memory allocation.	When the tool collects the information, it creates a memory dump file that can be read with any Java profiler tool.	For more information, see “memorydump” on page 23.
The memorydump diagnostic tool cannot start or analyze information.	Before you start the tool, make sure that the server instance is running. The tool requires the mws.pid file that is available in the server temp directory on the file system. The system creates this file when the server instance is started and removes the file when the server instance is shut down.	For more information, see: <ul style="list-style-type: none"> ■ “memorydump” on page 23. ■ “Using the Memory Monitor Portlet” on page 52.
The output file of the memory diagnostic tool is too big to be handled easily.	If the created file is large, the tool enables you to compress it and create a smaller .zip file with the compress option.	For more information, see “memorydump” on page 23.
You want to analyze the report file that is created by the memory diagnostic tool.	The memorydump tool captures the required information and stores it in a memory dump file. You can open and analyze the memory dump file with the standard JDK 1.6 jVisualVM tool	For more information, see: <ul style="list-style-type: none"> ■ “envcapture” on page 14 ■ “memorydump” on page 23.

Problem	Description	Reference
	that is available in the JAVA_home/bin directory, or an alternative Java profiler tool.	<ul style="list-style-type: none"> ■ “Using External Command-Line Diagnostic Tools” on page 33.
You receive many e-mail notifications about reaching memory thresholds. In some cases, the server restarts frequently.	System administrators configure the Memory Monitor properties and set the memory thresholds that define memory notifications, and, if configured, enable server restarts.	Adjust the memory thresholds. For more information, see “Using the Memory Monitor Portlet” on page 52.
You do not receive e-mail notifications when the memory thresholds are reached.	Memory Monitor notifications are sent only to the recipient e-mail accounts defined in the Memory Monitor configuration.	Make sure all intended recipients are listed in the configuration. For more information, see “Using the Memory Monitor Portlet” on page 52.
The server instance is not restarted after the FATAL memory threshold is reached.	The restartServer field is set to <code>false</code> by default.	Enable server restart upon FATAL conditions. For more information, see “Using the Memory Monitor Portlet” on page 52.
The tool does not analyze and display memory-monitoring information.	Configure the Memory Monitor properties and activate the tool by entering <code>true</code> in the enabled text field.	For more information, see “Using the Memory Monitor Portlet” on page 52.
The Memory Monitor is active but does not display information.	In My webMethods, enable the display by starting the memory monitor service.	For more information, see “Using the Memory Monitor Portlet” on page 52.
You want to monitor and analyze the performance of your CAF application.	The Performance Analysis tool enables portlet developers or administrators to measure and analyze the performance of My webMethods Server services and custom portlets or applications using an embedded performance-monitoring service.	For more information, see “Using the Performance Analysis Portlet” on page 39.
You want to analyze the application performance on alternative systems.	You can export the performance data and later on provide it for further analysis or import it and analyze it on another system or save it as reference information.	For more information, see “Using the Performance Analysis Portlet” on page 39.

Problem	Description	Reference
The performance information is too comprehensive or too superficial.	Depending on the service that you configure and use, the tool displays lightweight or comprehensive performance information.	For more information, see “Using the Performance Analysis Portlet” on page 39.
The performance information is not detailed enough	If you own the source code of the application being tested you can implement custom performance tokens to capture more information.	Add performance tokens and enable the tool for comprehensive reporting. For more information, see “Using the Performance Analysis Portlet” on page 39.
You are not satisfied with the displayed content of a diagnostic portlet.	Configure the displayed content of My webMethods Server diagnostic portlet.	For more information, see: <ul style="list-style-type: none"> <li data-bbox="1068 751 1446 814">■ “Using the Thread Dump Portlet” on page 38. <li data-bbox="1068 846 1474 909">■ “Using the Memory Monitor Portlet” on page 52. <li data-bbox="1068 940 1430 1039">■ “Using the Performance Analysis Portlet” on page 39. <li data-bbox="1068 1071 1430 1169">■ “Using the Performance Statistics Portlet” on page 43. <li data-bbox="1068 1201 1430 1262">■ “Using the Log Analysis Tools” on page 46.
You cannot import previously exported server snapshot information.	The tool imports an XML file and if the file structure contains performance information with the expected format, the tool analyzes and displays the snapshot information. The tool also exports a snapshot in CSV format, but this file cannot be imported and analyzed.	For more information, see “Using the Performance Analysis Portlet” on page 39.
The Log Analysis tool you are working with is not processing the correct log file.	The targeted log files (data source) is not configured correctly.	For more information, see “Configuring Log Analysis Data Sources” on page 46.

Problem	Description	Reference
The Log Analysis tool you are working with does not display correct entries on a date basis.	Be sure your date ranges are correctly set. If you select a period that goes beyond the captured date range, the log data is displayed only for what is captured in the logs.	For more information, see “Configuring the Log Analysis Time Period” on page 47.

Troubleshooting Environment Issues

My webMethods Server has a set of tools that enable you to easily identify and troubleshoot possible issues that you may encounter while working with it. This section presents some common configuration problems that are addressed by the diagnostic tools. The tools facilitate collecting the needed troubleshooting data and additionally provide output files that can be used by a support organization, such as Software AG Global Support or your internal support group.

The following table outlines potential problems, a description of the problem, and a reference to the tool that assists in the resolution:

Problem	Description	Reference
You need external support for your My webMethods Server.	When you request assistance from the Software AG Global Support, you are asked to provide additional environment-specific information for My webMethods Server.	For more information, see “envcapture” on page 14.
You need to report information about: <ul style="list-style-type: none"> ■ Log files. ■ Configuration files. ■ Installation folder structure, size, time stamps. ■ File system permissions for the current operating system user account. ■ Details about the local operating system and Java virtual machine. ■ Details about the Internet connectivity and the 	Use the envcapture command line tool to capture the needed information for your My webMethods Server installation.	For more information, see “envcapture” on page 14.

Problem	Description	Reference
connectivity to My webMethods Server database.		
The report file created by the envcapture tool does not contain the needed information.	Use the available <code>-a --action</code> command option to configure the envcapture tool to report the information that you want to capture.	For more information, see “envcapture” on page 14 .
You want to compare snapshots created by the envcapture command line tool.	You can use the envdiff tool to compare XML files generated by the envcapture tool.	For more information, see “envdiff” on page 17 .
You encounter issues related to environment changes.	You can identify and troubleshoot issues related to environment changes.	For more information, see “envdiff” on page 17 .
Outline the differences in the following attributes of XML snapshot files: <ul style="list-style-type: none"> ■ Folder structure ■ File size ■ MD5 hash ■ Timestamp ■ Permissions 	You can use the envdiff tool to examine the structure of the XML files and identify differences in the corresponding environment attributes.	For more information, see “envdiff” on page 17 .
You need to identify memory leaks within My webMethods Server.	Using the diagnostic tools you can identify possible memory leaks or troubleshoot memory allocation problems.	For more information, see: <ul style="list-style-type: none"> ■ “memorydump” on page 23. ■ “Using the Memory Monitor Portlet” on page 52.
Request further assistance from an external support organization about the memory allocation.	Use the memorydump tool to collect information and create a memory dump file.	For more information, see “memorydump” on page 23 .
You want to capture memory allocation information on another server instance.	The <code>-s --server</code> option enables you to capture memory allocation information for other server instances on the local system.	For more information, see “memorydump” on page 23 .

Troubleshooting Performance Issues

My webMethods Server has a set of tools that enable you to easily identify and troubleshoot possible issues that you may encounter while working with it. This section presents some common configuration problems that are addressed by the diagnostic tools. The tools facilitate collecting the needed troubleshooting data and additionally provide output files that can be used by a support organization, such as Software AG Global Support or your internal support group.

The following table outlines potential problems, a description of the problem, and a reference to the tool that assists in the resolution:

Problem	Description	Reference
My webMethods Server exhibits slow performance, stops responding, or encounters a deadlock condition.	Use the <code>threaddump</code> tool and portlet to establish a remote connection to the server and then to obtain thread dump information.	For more information, see: <ul style="list-style-type: none"> ■ “threaddump” on page 25. ■ “Using the Thread Dump Portlet” on page 38.
My webMethods Server exhibits slow performance or stops responding until other processes are finished.	Additional manual thread dump analysis is necessary. To do so, the tool displays all available threads. If the tool does not display deadlock threads, configure the settings to display the complete thread dump information and use the results for further analysis.	For more information, see: <ul style="list-style-type: none"> ■ “threaddump” on page 25. ■ “Using the Thread Dump Portlet” on page 38.
My webMethods Server stops responding and you cannot diagnose it using the portlet diagnostic tool.	Use the command line tool, which relies on a JVM algorithm to diagnose threads.	For more information, see “threaddump” on page 25.
You experience problems regarding the memory allocation of My webMethods Server.	Use the <code>memorydump</code> command line tool or the Memory Monitor portlet to troubleshoot the memory allocation of My webMethods Server.	For more information, see: <ul style="list-style-type: none"> ■ “memorydump” on page 23. ■ “Using the Memory Monitor Portlet” on page 52.
You want to monitor and analyze the performance of your CAF application.	The Performance Analysis portlet enables developers or administrators to measure and analyze the performance of My webMethods Server services and custom portlets or applications using an embedded	For more information, see “Using the Performance Analysis Portlet” on page 39.

Problem	Description	Reference
	performance-monitoring service.	
You want to analyze application performance on other systems.	You can export the performance data and then import it and analyze it on another system, or save it as reference information.	For more information, see “Using the Performance Analysis Portlet” on page 39.
The performance information is too comprehensive or too superficial.	Depending on the service that you configure and use, the tool displays lightweight or thorough performance information.	For more information, see “Using the Performance Analysis Portlet” on page 39.
You want to analyze a snapshot within a particular period.	The Performance Analysis tool enables you to capture a particular snapshot and then analyze the performance within the particular interval.	For more information, see “Using the Performance Analysis Portlet” on page 39.
You need performance support from an external organization.	The Performance Analysis tool enables you to export the snapshot into an XML file and provide the exported data for further analysis. The tool also enables you to export the analyzed data into a comma separated values (CSV) file. Note that you cannot import and analyze CSV files.	For more information, see “Using the Performance Analysis Portlet” on page 39.
You want to analyze the overall performance.	As an administrator, you can use the Performance Analysis diagnostic tool to analyze overall performance.	For more information, see “Using the Performance Analysis Portlet” on page 39.
The performance service settings are not persisted after a server restart.	The Performance Analysis tool configuration enables you to persist the performance service type and state configuration settings after server restart. If you do not configure this option, the changes are reset to the default configuration upon restart.	For more information, see “Using the Performance Analysis Portlet” on page 39.

Problem	Description	Reference
You cannot import previously exported server snapshot information.	The Performance Analysis tool cannot import a snapshot in CSV format .	For more information, see “Using the Performance Analysis Portlet” on page 39.
You want to define and troubleshoot the slowest database calls.	You can use a diagnostic tool to display information about ten of the slowest database calls.	For more information, see “Using the Performance Statistics Portlet” on page 43.
You want to define and troubleshoot the slowest performing portlets.	Use the Performance Statistics portlet to display performance information about the ten slowest portlets.	For more information, see “Using the Performance Statistics Portlet” on page 43.
You want to define and troubleshoot the slowest performing Web services.	Use the Performance Statistics portlet to display performance information about the ten slowest Web services.	For more information, see “Using the Performance Statistics Portlet” on page 43

Troubleshooting Log Issues

My webMethods Server provides a set of pre-configured log analysis tools that read and analyze log files created within My webMethods Server in accordance with the log4j mechanism. As an administrator, you configure a common data source file and analysis time period which is then used by all log analysis tools.

These tools are available in the My webMethods interface at **Tools > MWS Monitoring and Diagnostics > Log Tracing and Analysis**

The following table outlines potential problems, a description of the problem, and a reference to the tool that assists for the resolution:

Problem	Description	Reference
You want to analyze the errors.log file.	The loganalyzer diagnostic tool analyzes the errors.log file, which is an XML file created in accordance with the log4j mechanism. The tool cannot examine plain text log files.	For more information, see “loganalyzer” on page 19.
You want to organize the reported log issues on My webMethods Server.	The loganalyzer tool examines the errors.log file you provide for analysis, and organizes the results in several reports that are displayed in the console window.	For more information, see “loganalyzer” on page 19.

Problem	Description	Reference
	Depending on the options you use when you start the tool, you can create several XML files that contain different reports.	
You want to analyze the average amount of log issues.	Use the loganalyzer diagnostic tool to quickly identify and analyze the amount of reported issues in the logs of My webMethods Server.	For more information, see “loganalyzer” on page 19.
The loganalyzer tool cannot map the issues to the relevant Java classes or packages.	The javaArtifactMapping.xml file might be out -of-date and may not contain information about new custom components.	For more information, see “Java Artifact Mapping in My webMethods Server ” on page 31.
The Log Analysis tool you are working with is not processing the correct log file.	The targeted log files (data source) is not configured correctly.	For more information, see “Configuring Log Analysis Data Sources” on page 46.
The Log Analysis tool you are working with does not display correct entries on a date basis.	Be sure your date ranges are correctly set. If you select a period that goes beyond the captured date range, the log data is displayed only for what is captured in the logs.	For more information, see “Configuring the Log Analysis Time Period” on page 47.
You want to display statistics for all log issues.	The Log Analysis tools display statistics about the logs in the configured data source file that occur in the configured period. Information about the total number of appearances of the issues is displayed, as well as the time interval in which the issues appeared.	For more information, see “Using the Predefined Logging Tools” on page 49.
You want to display and investigate only FATAL issues.	The Log FATAL Issues tool analyzes and displays only log messages of FATAL severity.	For more information, see “Using the Predefined Logging Tools” on page 49
You want to display and investigate log issues by a specific component or product.	To display thorough information about a particular component type or product, choose the respective graphic that is displayed on the tools. Depending on your selection, the tool opens the Issues For	For more information, see “Using the Predefined Logging Tools” on page 49

Problem	Description	Reference
	Component or Issues For Product tools and displays information about the logs of the respective component or product.	
You want to find out which portlets are used the most.	The Most Used Portlets tool displays statistics about the ten most-used portlets in My webMethods Server.	For more information, see “Using the Predefined Logging Tools” on page 49
You want to find out which services are used the most	The Most Used Web Services tool displays statistics about the ten most-used Web services in My webMethods Server.	For more information, see “Using the Predefined Logging Tools” on page 49
You want to obtain more information about certain issues that are displayed by the other diagnostic tools.	The Log Issues tool analyzes and displays only log messages with the following severity: <ul style="list-style-type: none"> ■ WARNING ■ ERROR ■ FATAL 	For more information, see “Using the Predefined Logging Tools” on page 49
You want to display more information about issues that are displayed by the Log Issues portlet.	The Issue Details window displays thorough information about a particular log issue. You open the window by clicking a log issue in the following tools: <ul style="list-style-type: none"> ■ Log FATAL Issues ■ Log Issues 	For more information, see “Using the Predefined Logging Tools” on page 49

A JMX MBean Attributes and Actions

■ Categories under Root	72
■ Categories under Core	74

Categories under Root

This section contains JMX MBean attributes and actions for the root category of `com.softwareag.mws`.

Type Cache

The following table lists the categories of type Cache:

Categories of Type Cache

Access	A cache for a specified user's rights.
Alias	A cache for alias keys.
Bpm Process Definitions	A cache for BPM Process Definitions. A transient cache.
Bpm Process Instance	A cache for BPM Process Instances. A transient cache.
Container	A cache for refresh xtype containers. A transient cache.
Group	A cache for storing the groups of which a principal is a member.
Portlet Transient	A cache of objects used by a portlet. A transient cache.
Presentation	A cache of rendered HTML fragments.
Role	A cache for storing the roles of which a principal is a member.
TaskData	A cache for tasks data. A transient cache.
Thing	A cache for My webMethods Server's ThingIDs. A transient cache.
ThingRelation	A cache for ThingID relations to objects. A transient cache.
Transient cache for Alias	A transient cache version of the Alias cache.
Transient cache for Presentation	A transient cache version of the Presentation cache.

The following table lists the attributes for type Cache:

Attributes for type cache

cacheID	The return ID of the cache.
cacheSize	The number of items in the cache.
uniqueKeys	The number of unique keys for the cache.
maxSize	The maximum number of items in the cache.
validEntries	Items that resolve to objects.

Attributes for type cache

invalidEntries	Items that do not resolvable to objects.
expiredEntries	Items that have reached the expiration time.
expirableEntries	Items that have an expiration time.
cacheHits	Items found in the cache.
cacheMisses	Items not found in the cache.
totalDependencies	The total count of entries that have dependencies.
entriesWithDependencies	The count of entries with dependencies to other items in the cache.

The following table lists the actions, applicable to type Cache:

Actions for type cache

resetCounters	Zero out the count of cache hits and misses.
---------------	----------------------------------------------

Type Connection Pool

The following table lists the categories of type Connection Pool:

Categories of type connection pool

default	The default connection pool.
---------	------------------------------

The following table lists the attributes for type Connection Pool:

Attributes for type connection pool

checkinCalls	Check an object into this pool.
checkoutCalls	Check out an object from this pool.
createCalls	A count of created objects added to this pool.
poolSize	Idle size plus checked out size.
idleSize	A count of idle objects in this pool.
checkedOutSize	A count of checked out objects in this pool.
highWaterMark	A statistic count of the highest number objects in this pool.
maxSize	The maximum number of objects in this pool.
minSize	The minimum number of objects in this pool.

Attributes for type connection pool

initialSize	The initial number of objects in this pool.
closed	Indicates this pool is shut down.

The following table lists the actions, applicable to type Connection Pool:

Actions for type connection pool

None

Categories under Core

This section contains JMX MBean attributes and actions for the core category of `com.softwareag.mws`.

Type Events

The following table lists the attributes for type Events:

Attributes for type events

loginEvents	A count of User login events.
loginFailedEvents	A count of User failed login events.
logoutEvents	A count of User logout events.
createEvents	A count of new things created.
readEvents	A count of web GET requests.
updateEvents	A count of things modified.
deleteEvents	A count of things deleted.

The following table lists the actions, applicable to type Events:

Actions for type events

resetCounters	Zeros out all event counters.
---------------	-------------------------------

Type JMXPerf

Type JMXPerf is a low-level performance monitoring utility exposed to JMX. This type is applied to any My webMethods Server internal Java class.

The following table lists the categories of type JMXPerf:

Categories of Type JMXPerf

Context
 Database
 Directory
 Handler
 MWSInternal
 MWSProviderInt
 Manager
 Portlet
 RTL
 SOAP
 Task
 mws-rules
 ndlNodeSession
 ndlToolProvider
 webService

The following table lists the attributes for type JMXPerf:

Attributes for type JMXPerf

TotalTime (ms)	Total time for all the actions performed, in milliseconds.
numActions	The number of actions performed.
averageTime (ms)	Average time per action, in milliseconds.
peakTime (ms)	The largest time interval for an action, in milliseconds.

The following table lists the actions, applicable to type JMXPerf:

Actions for type JMXPerf

resetCounter	Zeros out the0 totaltime, numActions, and peakTime counters.
--------------	--------------------------------------------------------------

Type Logs

The following table lists the categories of type Logs:

Categories of type Logs — All log4j appenders configured for My webMethods Server and the OSGi platform

Framework	My webMethods Server framework.
shiro.AuthorizationServiceImpl	Platform security
DbJMSSClient	JMS Client

The following categories only have the total attribute

Logs.error	A cumulative count of all errors logged by all logs tracked by JMX.
Logs.fatal	A cumulative count of all fatal errors logged by all logs tracked by JMX.
Logs.warn	A cumulative count of all warnings logged by all logs tracked by JMX.

The following table lists the attributes for type Logs:

Attributes for type logs

fatal	A count of fatal events recorded in this log. See log4j Fatal.
error	A count of error events recorded in this log. See log4j Error.
warn	A count of warn events recorded in this log. See log4j Warn.

The following table lists the actions, applicable to type Logs:

Actions for type logs

resetCounters	Zeros out all logs counters.
---------------	------------------------------

Type Memory

The following table lists the attributes for type Memory:

Attributes for type memory

maxMemory (MB)	The maximum memory for the Java Virtual Machine running My webMethods Server, in megabytes.
freeMemory (MB)	The amount of free memory in the Java Virtual Machine running My webMethods Server, in megabytes.

The following table lists the actions, applicable to type Memory:

Actions for type memory

None

Type Sessions

The following table lists the attributes for type Sessions:

Attributes for type sessions

totalSessions	A count of all the sessions created, including active users and system sessions.
activeUsers	A count of authenticated users, excluding system sessions.

The following table lists the actions, applicable to type Sessions:

Actions for type sessions

None

Type tasks

The following table lists the attributes for type Tasks:

Attributes for type tasks

queueEvents	A count of new tasks events.
completedEvents	A count of completed tasks events.
errorEvents	A count of error tasks events.

The following table lists the actions, applicable to type Tasks:

Actions for type tasks

resetCounters	Zeros out all events tasks.
---------------	-----------------------------

Type Directory Services Cache

The following table lists the categories of type Directory Services Cache:

Categories of type directory services cache

DbDirServices82	The database directory services cache.
-----------------	----------------------------------------

Categories of type directory services cache

system The system directory services cache.

The following table lists the attributes for type Directory Services Cache:

Attributes for type directory services cache

dnCacheCapacity	DN (Distinguished Name) cache maximum size in entries.
dnCacheSize	DN (Distinguished Name) cache current size in entries.
uriCacheCapacity	URI cache maximum size in entries.
uriCacheSize	URI cache current size in entries.
queryCacheCapacity	Query cache maximum size in entries.
queryCacheSize	Query cache current size in entries.
authenticateCacheCapacity	Query cache maximum size in entries.
authenticateCacheSize	Query cache current size in entries.

The following table lists the actions, applicable to type Directory Services Cache:

Actions for type directory services cache

None

Type Task Change Handler Executor

The following table lists the attributes for type Task Change Handler Executor:

Attributes for type task change handler executor

active count	Active thread count. The approximate number of threads that are actively executing tasks.
--------------	-------------------------------------------------------------------------------------------

The following table lists the actions, applicable to type Task Change Handler Executor:

Actions for type task change handler executor

None