

Working with the webMethods Product Suite and the Java Service Wrapper

Version 9.7

October 2014

This document applies to webMethods Product Suite Version 9.7 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 © 2014 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 <http://documentation.softwareag.com/legal/>.

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 <http://documentation.softwareag.com/legal/> 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 and license terms, please refer to "License Texts, Copyright Notices and Disclaimers of Third Party Products". This document is part of the product documentation, located at <http://documentation.softwareag.com/legal/> and/or in the root installation directory of the licensed product(s).

Table of Contents

About this Guide	5
Document Conventions.....	5
Documentation Installation.....	6
Online Information.....	6
The Java Service Wrapper	7
What is the Java Service Wrapper?.....	8
Determining Which Version of the Java Service Wrapper Your webMethods Product Is Using.....	8
The Java Service Wrapper Configuration Files.....	9
Viewing and Editing the Java Service Wrapper Properties.....	9
JVM Configuration.....	10
JVM Configuration Properties.....	10
Memory Properties.....	10
Java Location Properties.....	11
Classpath Properties.....	11
Java Library Path Properties.....	11
Additional Properties.....	12
Override Properties.....	13
The Wrapper Log.....	14
Logging Properties.....	14
Fault Monitoring.....	15
Fault-Monitoring Properties.....	16
JVM Timeout Properties.....	16
Deadlock-Detection Properties.....	16
Console Filtering Properties.....	17
Generating a Thread Dump.....	17
Software AG Common Platform Profile Names	19

About this Guide

This guide provides an introduction to the ways in which the webMethods product suite uses the Java Service Wrapper. It describes property settings you use to configure the Java Service Wrapper and provides procedures for using utility commands provided by the wrapper.

Document Conventions

Convention	Description
Bold	Identifies elements on a screen.
Narrowfont	Identifies storage locations for services on webMethods Integration Server, using the convention <i>folder.subfolder:service</i> .
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
<i>Italic</i>	Identifies variables for which you must supply values specific to your own situation or environment. Identifies new terms the first time they occur in the text.
Monospace font	Identifies text you must type or messages displayed by the system.
{ }	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 (...).

Documentation Installation

You can download the product documentation using the Software AG Installer. The documentation is downloaded to a central directory named `_documentation` in the main installation directory (SoftwareAG by default).

Online Information

Software AG Documentation Website

You can find documentation on the Software AG Documentation website at <http://documentation.softwareag.com>. The site requires Empower credentials. If you do not have Empower credentials, you must use the TECHcommunity website.

Software AG Empower Product Support Website

You can find product information on the Software AG Empower Product Support website at <https://empower.softwareag.com>.

To submit feature/enhancement requests, get information about product availability, and download products and certified samples, go to [Products](#).

To get information about fixes and to read early warnings, technical papers, and knowledge base articles, go to the [Knowledge Center](#)

Software AG TECHcommunity

You can find documentation and other technical information on the Software AG TECHcommunity website at <http://techcommunity.softwareag.com>. You can:

- Access product documentation, if you have TECHcommunity credentials. If you do not, you will need to register and specify "Documentation" as an area of interest.
- Access articles, demos, and tutorials.
- Use the online discussion forums, moderated by Software AG professionals, to ask questions, discuss best practices, and learn how other customers are using Software AG technology.
- Link to external websites that discuss open standards and web technology.

1 The Java Service Wrapper

■ What is the Java Service Wrapper?	8
■ Determining Which Version of the Java Service Wrapper Your webMethods Product Is Using	8
■ The Java Service Wrapper Configuration Files	9
■ JVM Configuration	10
■ The Wrapper Log	14
■ Fault Monitoring	15
■ Generating a Thread Dump	17

What is the Java Service Wrapper?

The Java Service Wrapper is an application developed by Tanuki Software, Ltd. Some webMethods products use the Java Service Wrapper to:

- Start and stop the Java Virtual Machines (JVM) in which they run.
- Record the console output from the JVM in a log file.
- Monitor the JVM for various fault conditions and take a prescribed action when a fault occurs.
- Enable you to obtain a thread dump when the JVM is running as a service under Windows.

This guide discusses webMethods products that use the Java Service Wrapper and run on the Software AG Common Platform. There are some webMethods products that use the Java Service Wrapper but do not run on the Software AG Common Platform. For information about how these products use the wrapper, see their product documentation.

Determining Which Version of the Java Service Wrapper Your webMethods Product Is Using

The webMethods products running in your Software AG installation might use different versions of the Java Service Wrapper. Knowing the version number is helpful when you refer to the product documentation for the Java Service Wrapper on the Tanuki Software, Ltd. website. To determine which version of the wrapper a given product is using, use the following procedure.

To determine which version of the wrapper your webMethods product is using

1. From the command line on the machine where the webMethods product is installed, navigate to the following directory:

```
Software AG_directory\profiles\profile name \bin
```

Note: *profile name* is the name of the profile for your webMethods product. For a list of products and their corresponding profile names, see "[Software AG Common Platform Profile Names](#)" on page 19.

2. Execute the following command:

```
service -version
```


The Java Service Wrapper Configuration Files

webMethods products that run on the Software AG Common Platform each have a pair of configuration files for the Java Service Wrapper. The properties in these files determine the configuration of the JVM that the wrapper launches for the given product (i.e., heap size, Java system property settings, and so forth), and specify the behavior of the logging and monitoring features of the Java Service Wrapper.

File name	Description
wrapper.conf	<p>Contains property settings that are installed by the webMethods product.</p> <p>Important: Do not modify the contents of this file unless instructed to do so by Software AG.</p>
custom_wrapper.conf	<p>Contains properties that modify the original settings installed by the webMethods product. Settings in this file override the settings in wrapper.conf.</p> <p>If you need to modify the property settings for the Java Service Wrapper, make your changes in this file.</p>

Important: The webMethods products that use the Java Service Wrapper have different policies regarding which properties you can configure. Before changing any property settings in the configuration files, review the Java Service Wrapper section in the administrator's guide for your particular webMethods product to determine which configuration changes it supports.

These configuration files are located in the following directory:

Software AG_directory\profiles*profile name* \configuration

Note: *profile name* is the name of the profile for your webMethods product. For a list of products and their corresponding profile names, see "[Software AG Common Platform Profile Names](#)" on page 19.

Viewing and Editing the Java Service Wrapper Properties

Use the following procedure to view or edit the configuration properties for the Java Service Wrapper.

To view or edit the Java Service Wrapper properties

1. Open the following configuration files in a text editor:

Software AG_directory\profiles\profile name \wrapper.conf

Software AG_directory\profiles\profile name \custom_wrapper.conf

2. Examine the properties in both files. In cases where a property exists in both files, the property in custom_wrapper.conf overrides the one in wrapper.conf.
3. If you need to modify a property, do the following:
 - a. Add the property to custom_wrapper.conf if it does not already exist in that file.
 - b. Edit the property setting in custom_wrapper.conf. *Do not make any changes to wrapper.conf.*
4. If you made changes to custom_wrapper.conf, save the file.
5. Close the wrapper.conf file without saving it.
6. Restart the webMethods product.

Important: When working with sequenced attribute properties, make sure that the sequence of properties in custom_wrapper.conf matches the sequence of properties in wrapper.conf.

JVM Configuration

The Java Service Wrapper configuration files include properties for specifying Java startup parameters (for example, heap size, PermGen size, and classpath) and for passing Java system properties into the JVM (that is, any property you would pass to the JVM using the -D option on the command line).

The Java Service Wrapper uses these properties to configure the JVM that it launches for a given webMethods product.

JVM Configuration Properties

You can use the following properties to configure the JVM that the Java Service Wrapper launches. To modify these properties, edit the custom_wrapper.conf file using the procedure in ["Viewing and Editing the Java Service Wrapper Properties" on page 9](#).

For additional details about these parameters, refer to the product documentation on the Tanuki Software, Ltd. website.

Important: Review the Java Service Wrapper section in the administrator's guide for your particular webMethods product to determine which of the following JVM configuration properties you can configure.

Memory Properties

Use the following properties to specify size of the Java heap.

<u>This property...</u>	<u>Specifies...</u>
<code>wrapper.java.initmemory</code>	The initial size (in MB) of the Java heap.
<code>wrapper.java.maxmemory</code>	The maximum size (in MB) to which the Java heap can grow.

Java Location Properties

Use the `wrapper.java.command` property to specify the Java executable file.

<u>Property</u>	<u>Specifies</u>
<code>wrapper.java.command</code>	The fully qualified path to the Java executable file.

Classpath Properties

Use `wrapper.java.classpath` properties to specify jars or directories in the classpath.

<u>Property</u>	<u>Specifies</u>
<code>wrapper.java.classpath.n</code>	<p>A jar or directory in the classpath. Each <code>wrapper.java.classpath</code> property requires a unique sequence number, <i>n</i>, at the end of the property name. The sequence numbers specify the order of the jars or directories in the classpath.</p> <p>For example, the following classpath properties define a classpath made up of three directories:</p> <pre>wrapper.java.classpath.1=%myRoot%wrapper.jar wrapper.java.classpath.2=%myRoot%myApp.jar wrapper.java.classpath.3=%myRoot%myCommon.jar</pre> <p>To add another directory to this classpath, you would add a <code>wrapper.java.classpath.4</code> property to the end of this list.</p>

Java Library Path Properties

Use `wrapper.java.library.path` properties to set the paths for Java Native libraries.

Property	Specifies
<code>wrapper.java.library.path.n</code>	<p>A Java option to specify the path of the Java Native library. This is required if your product needs to access any OS specific libraries.</p> <p>Each <code>wrapper.java.library.path</code> property requires a unique sequence number, <i>n</i>, at the end of the property name.</p> <p>In the following example, the <code>wrapper.java.library.path</code> property is used to add three libraries to the classpath.</p> <pre>wrapper.java.library.path.14= C:\SoftwareAG\IntegrationServer\instances\default\lib wrapper.java.library.path.15= C:\SoftwareAG\IntegrationServer\instances\ default\packages\WmWin32\code\libs wrapper.java.library.path.16= C:\SoftwareAG\IntegrationServer\instances\default\ support\win32</pre>

Additional Properties

Use `wrapper.java.additional` properties to set additional Java parameters to pass to the JVM when it is launched.

Property	Specifies
<code>wrapper.java.additional.n</code>	<p>A Java option to be passed in on the command line.</p> <p>Each <code>wrapper.java.additional</code> property requires a unique sequence number, <i>n</i>, at the end of the property name. The sequence numbers specify the order in which the options will appear in the command line.</p> <p>In the following example, the <code>wrapper.java.additional</code> property is used to add three options to the <code>java</code> command line.</p> <pre>wrapper.java.additional.1=-Declipse.ignoreApp=true wrapper.java.additional.2=-Dosgi.noShutdown=true wrapper.java.additional.3=-XX:MaxPermSize=256M</pre> <p>To add another option to the command line, you would add a <code>wrapper.java.additional.4</code> property to the end of the list.</p>

Override Properties

Use `wrapper.app.parameter` properties pass command line switches and arguments to the product during startup.

Note: Currently, you can only use this feature with Integration Server. For more information, see *webMethods Integration Server Administrator's Guide*.

Property	Specifies
<code>wrapper.app.parameter.n</code>	<p>An option to provide command line switch options to the product. For example, in Integration Server you can use this property to pass switches such as <code>-log</code>, <code>-debug</code>, and so on.</p> <p>Each <code>wrapper.app.parameter</code> property requires a unique sequence number, <i>n</i>, at the end of the property name to specify the order in which Integration Server should activate the switches.</p> <p>Each <code>wrapper.app.parameter</code> property can contain only a single value. For switches that require a switch value (for example, <code>-port</code>), you must add two <code>wrapper.app.parameter</code> properties: One to specify the switch, and another to specify the value of the switch. These properties must be sequential. For example, to override the Integration Server primary port, you could define the <code>wrapper.app.parameter</code> properties as follows:</p> <pre>wrapper.app.parameter.9=-port wrapper.app.parameter.10=6666</pre> <p>For more information about using switches to override settings in Integration Server, see <i>webMethods Integration Server Administrator's Guide</i>.</p> <p>For every <code>wrapper.app.parameter</code> property you add to <code>custom_wrapper.conf</code>, you must also modify the <code>wrapper.app.parameter.2</code> property to reflect the total number of <code>wrapper.app.parameter</code> properties that will be modified for the JVM. For example, if the <code>wrapper.app.parameter.2</code> property is set to 4 (the default) and you add two <code>wrapper.app.parameter</code> properties to override the Integration Server primary port, you would increase the value of <code>wrapper.app.parameter.2</code> by two. After making your edits, the <code>wrapper.app.parameter.2</code> property would appear as follows:</p>

Property**Specifies**

```
wrapper.app.parameter.2=6
```

The Wrapper Log

The Java Service Wrapper writes the console output from the JVM to a log file. This log includes stack traces that the JVM produces when a process throws an exception. It also includes any thread dumps you generate from the JVM. The wrapper log is particularly useful when a webMethods product runs as a Windows service, because console output is not normally available to you in this mode.

The log resides in the following directory:

```
Software AG_directory\profiles\profile name \logs\wrapper.log
```

Note: *profile name* is the name of the profile for your webMethods product. For a list of products and their corresponding profile names, see "[Software AG Common Platform Profile Names](#)" on page 19.

Logging Properties

The following properties are used to configure the content, format, and behavior of the wrapper log. These properties include settings to limit the size of the log file and specify how many old log files to retain. To modify these properties, edit the `custom_wrapper.conf` file using the procedure in "[Viewing and Editing the Java Service Wrapper Properties](#)" on page 9.

For additional details about these parameters, refer to the product documentation for the Java Service Wrapper on the Tanuki Software, Ltd. website.

Important: Review the Java Service Wrapper section in the administrator's guide for your particular webMethods product to determine which of the following logging properties you can configure.

This property...	Specifies...
<code>wrapper.console.loglevel</code>	The level of detail displayed in the console.
<code>wrapper.console.format</code>	The format of the messages displayed in the console.
<code>wrapper.logfile</code>	The file into which console output for Integration Server is logged.

<u>This property...</u>	<u>Specifies...</u>
<code>wrapper.logfile.loglevel</code>	The level of detail recorded in the wrapper log file. This setting must be a level that is equal to or lower than the setting in <code>wrapper.console.loglevel</code> .
<code>wrapper.logfile.format</code>	The format of the messages recorded in the wrapper log file.
<code>wrapper.logfile.maxsize</code>	The maximum size to which the log can grow.
<code>wrapper.logfile.maxfiles</code>	The number of old logs the Java Service Wrapper retains.
<code>wrapper.syslog.loglevel</code>	Which messages are written to the Event Log on Windows systems, or the syslog on UNIX systems.

Fault Monitoring

The Java Service Wrapper can monitor the JVM and restart the JVM or perform other actions when it detects specified fault conditions. The fault-monitoring features the wrapper provides are:

- **Detecting a nonoperational (i.e., hung) JVM.** After the Java Service Wrapper starts the JVM, it pings the JVM periodically to check whether it is operational. If the JVM does not respond to a ping within a specified interval, the Java Service Wrapper assumes that the JVM has stopped functioning and restarts it.

Properties in the Java Service Wrapper configuration files determine the ping interval and timeout period. Each webMethods product configures these properties differently. Certain products disable this feature entirely.

- **Detecting thread deadlocks in the JVM.** A thread deadlock occurs when two or more threads attempt to lock resources in a manner that causes all threads to wait indefinitely. The Java Service Wrapper can monitor the JVM for a deadlock condition and take a specified action (for example, restarting the JVM) when the condition occurs.

Properties in the Java Service Wrapper configuration files determine whether the deadlock detection is enabled, how often it checks for deadlock threads, and what action to take when a deadlock occurs. Most webMethods products disable this feature by default.

- **Detecting specified messages in the console output.** The Java Service Wrapper can monitor the console output and take a specified action when a given string of text appears. This feature is often used to watch for out-of-memory messages.

Properties in the Java Service Wrapper configuration files specify the string of text to watch for and the action to take when the string is detected.

Fault-Monitoring Properties

The following properties are used to configure the fault-monitoring features in the Java Service Wrapper. To modify these properties, edit the `custom_wrapper.conf` file using the procedure in ["Viewing and Editing the Java Service Wrapper Properties"](#) on page 9.

For additional details about these parameters, refer to the product documentation for the Java Service Wrapper on the Tanuki Software, Ltd. website.

Important: Review the Java Service Wrapper section in the administrator's guide for your particular webMethods product to determine which of the following properties you can configure.

JVM Timeout Properties

Property	Specifies
<code>wrapper.ping.interval</code>	How often, in seconds, the Java Service Wrapper pings the JVM to determine whether it is active.
<code>wrapper.ping.timeout</code>	The length of time, in seconds, that the Java Service Wrapper waits for a response to a ping. If it does not receive a response in the specified time, it initiates the action specified in <code>wrapper.ping.timeout.action</code> .
<code>wrapper.ping.timeout.action</code>	The action that occurs if the Java Service Wrapper does not receive a response to a ping in the allotted time.

Deadlock-Detection Properties

Property	Specifies
<code>wrapper.check.deadlock</code>	Whether Java Service Wrapper monitors the JVM for deadlocked threads. For most products, this feature is disabled by default.
<code>wrapper.check.deadlock.interval</code>	How often, in seconds, the Java Service Wrapper evaluates the JVM for a deadlock condition.

Property	Specifies
<code>wrapper.check.deadlock.action</code>	Action that occurs if the Java Service Wrapper detects a deadlock condition.

Console Filtering Properties

Property	Specifies
<code>wrapper.filter.trigger.n</code>	A string of text that you want to detect in the console output.
<code>wrapper.filter.action.n</code>	The action that occurs when the Java Service Wrapper detects the text specified in <code>wrapper.filter.trigger.n</code>
<code>wrapper.filter.allow_wildcards.n</code>	Whether the Java Service Wrapper processes wildcard characters that appear in the string of text specified in <code>wrapper.filter.trigger.n</code> .
<code>wrapper.filter.message.n</code>	The message that the Java Service Wrapper sends to the console when it detects the string of text specified in <code>wrapper.filter.trigger.n</code> .

Generating a Thread Dump

A thread dump can help you locate thread contention issues that can cause thread blocks or deadlocks. The Java Service Wrapper includes a utility that enables you to generate a thread dump for a webMethods product that is running as a Windows service.

To generate a thread dump of a webMethods product running as a Windows service

1. From the command line on the machine where the webMethods product is installed, navigate to the following directory:

```
Software AG_directory\profiles\profile name \bin
```

Note: *profile name* is the name of the profile for your webMethods product. For a list of products and their corresponding profile names, see "[Software AG Common Platform Profile Names](#)" on page 19.

2. Execute the following command:

```
service -dump
```

The Java Service Wrapper writes the thread dump to the wrapper log at
Software AG_directory\profiles*profile name* \logs\wrapper.log

A Software AG Common Platform Profile Names

The following are the profile names for webMethods products that run on the Software AG Common Platform.

Profile name	Product
appmondc	Application Monitor Data Collector
CCE	Software AG Command Central
CTP	Software AG Runtime
InfraDC	webMethods Infrastructure Data Collector
IS	webMethods Integration Server
MWS_ <i>instanceName</i>	My webMethods Server The profiles folder will contain one MWS_ <i>instanceName</i> directory for each My webMethods Server instance you have created.
ODE	webMethods OneData
SPM	Software AG Platform Manager