Working with the webMethods Product Suite and the Java Service Wrapper

Version 9.6

April 2014

WEBMETHODS

Table of Contents

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 *folder.subfolder:service*.
UPPERCASE	Identifies keyboard keys. Keys you must press simultaneously are joined with a plus sign (+).
Italic	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

You can find additional information about Software AG products at the locations listed below.

If you want to...	Go to...
Access the latest version of product documentation.	Software AG Documentation website http://documentation.softwareag.com
Find information about product releases and tools that you can use to resolve problems. See the Knowledge Center to: ■ Read technical articles and papers. ■ Download fixes and service packs (9.0 SP1 and earlier). ■ Learn about critical alerts. See the Products area to: ■ Download products. ■ Download certified samples. ■ Get information about product availability. ■ Access older versions of product documentation. ■ Submit feature/enhancement requests.	Empower Product Support website https://empower.softwareag.com

If you want to...	Go to...
■ Access additional articles, demos, and tutorials.	**Software AG Developer Community for webMethods**
■ Obtain technical information, useful resources, and online discussion forums, moderated by Software AG professionals, to help you do more with Software AG technology.	http://communities.softwareag.com/
■ Use the online discussion forums to exchange best practices and chat with other experts.	
■ Expand your knowledge about product documentation, code samples, articles, online seminars, and tutorials.	
■ Link to external websites that discuss open standards and many web technology topics.	
■ See how other customers are streamlining their operations with technology from Software AG.	

1 The Java Service Wrapper

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. To determine which version of the wrapper a given product is using, use the following procedure. (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 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 17.

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

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 17.

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 10.

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.

Important! *If you are using the webMethods Integration Server, do not modify any of the following settings.* Integration Server uses a script file to set the JVM configuration properties. To set the JVM configuration properties on webMethods Integration Server, see the section on configuring the JVM in the *webMethods Integration Server Administrator's Guide*.

Memory Properties

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

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

Classpath Properties

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

Property	Specifies
`wrapper.java.classpath.`n	A jar or directory in the classpath. Each `wrapper.java.classpath` property requires a unique sequence number, n, at the end of the property name. The sequence numbers specify the order of the jars or directories in the classpath.
	For example, the following classpath properties define a classpath made up of three directories:

	`wrapper.java.classpath.1=%myRoot%wrapper.jar`
	`wrapper.java.classpath.2=%myRoot%myApp.jar`
	`wrapper.java.classpath.3=%myRoot%myCommon.jar`

	To add another directory to this classpath, you would add a `wrapper.java.classpath.4` property to the end of this list.

Additional Properties

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

Property	Specifies
`wrapper.java.additional.`*n*	A Java option to be passed in on the command line.
	Each `wrapper.java.additional` property requires a unique sequence number, *n*, at the end of the property name. The sequence numbers specify the order in which the options will appear in the command line.
	In the following example, the `wrapper.java.additional` property is used to add three options to the `java` command line.

	`wrapper.java.additional.1=-Declipse.ignoreApp=true`
	`wrapper.java.additional.2=-Dosgi.noShutdown=true`
	`wrapper.java.additional.3=-XX:MaxPermSize=256M`

	To add another option to the command line, you would add a `wrapper.java.additional.4` property to the end of the list.

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 17.

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 10.

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...
`wrapper.console.loglevel`	The level of detail displayed in the console.
`wrapper.console.format`	The format of the messages displayed in the console.
`wrapper.logfile`	The file into which console output for Integration Server is logged.
`wrapper.logfile.loglevel`	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 `wrapper.console.loglevel`.
`wrapper.logfile.format`	The format of the messages recorded in the wrapper log file.
`wrapper.logfile.maxsize`	The maximum size to which the log can grow.
`wrapper.logfile.maxfiles`	The number of old logs the Java Service Wrapper retains.
`wrapper.syslog.loglevel`	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 theses 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 10.

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
wrapper.ping.interval	How often, in seconds, the Java Service Wrapper pings the JVM to determine whether it is active.
wrapper.ping.timeout	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 wrapper.ping.timeout.action.
wrapper.ping.timeout.action	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
wrapper.check.deadlock	Whether Java Service Wrapper monitors the JVM for deadlocked threads. For most products, this feature is disabled by default.

Property	Specifies
`wrapper.check.deadlock.interval`	How often, in seconds, the Java Service Wrapper evaluates the JVM for a deadlock condition.
`wrapper.check.deadlock.`	The action that occurs when the Java Service Wrapper detects a deadlock condition.

Console Filtering Properties

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

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 Window 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 .

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	webMethods Command Central
CTP	Software AG Runtime
InfraDC	webMethods Infrastructure Data Collector
IS	webMethods Integration Server
MWS_*instanceName*	My webMethods Server
	The profiles folder will contain one MWS_*instanceName* directory for each My webMethods Server instance you have created.
ODE	webMethods OneData
SPM	webMethods Platform Manager