

Symphony Marimba Reference Guide



Supporting
Symphony Marimba 9.0.00

November 2015

© Copyright 2015 Symphony Teleca, Corporation or its subsidiaries. All rights reserved. All information contained in this document is confidential and proprietary to Symphony Teleca, Corporation and may not be disclosed, reproduced, used, modified, made available, used to create derivative works, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, by or to any person or entity without the express written authorization of Symphony Teleca, Corporation. In consideration for receipt of this document, the recipient agrees to treat this document and its contents as confidential and agrees to fully comply with this notice. This document refers to numerous products by their trade names. In most, if not all, cases their respective companies claim these designations as Trademarks or Registered Trademarks. This document and the related software described herein are supplied under license agreement or nondisclosure agreement and may be used or copied only in accordance with the terms of such agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Symphony Teleca, Corporation. Contact Symphony Teleca, Corporation Customer Support to verify the date of the latest version of this document. The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental. Symphony Teleca, Corporation reserves all copyrights, trademarks, patent rights, trade secrets and all other intellectual property rights in this document, its contents and the software described herein.

Contacting Symphony Teleca Customer Support

You can obtain technical support by contacting Customer Support by telephone or e-mail. We are available 24/7/365:

Please send an e-mail to: CustomerSupport@Symphonyteleca.com OR

Call us at +1 214 396 0493 or US Toll Free Number +1 855 394 1543

Before contacting Symphony Teleca support

Please gather the following information and have it ready before contacting Symphony Teleca.

This will help us service your request immediately:

Marimba channel version for each module being used

Sequence of events leading to the issue

Error messages received along with the time and date that you received them

Environment details (number of transmitters, type of transmitters, number of endpoints, Operating System from servers and endpoints, Database version)

Details about the problem

Screenshots of errors

Attachments of relevant logs and configuration files





Contents

Preface	11
How to read this reference	11
Related documentation	12
Product guides	12
General guides	14
Using the documentation channel	14
Using the CMS Console to install the documentation channel	15
Using Channel Manager to install the documentation channel	15
Searching across the Symphony Marimba Client Automation documentation set	
15	
Using the Marimba Channel Store.	15
Locating the product documentation	16
Chapter 1 Command-line options	17
Overview	18
Command syntax.	18
The runchannel program	20
The tuner program	25
Application Packager options	34
Custom Application Packager command-line options	35
File Packager command-line options.	35
Packager for Shrinkwrap Windows Applications command-line options.	38
Windows Installer Packager command-line options	40
.NET Packager command-line options	41
PDA Packager command-line options	45

Virtual Packager command-line options	49
Upgrading channels	50
Checking the version of Application Packager and channels	51
Applying an XML template file to channels	51
Starting packaged applications from the command line	52
Staging MSI applications on endpoints	54
Certificate Manager options	54
Using runchannel for local certificate operations.	55
Using tuner -start for remote certificate operations.	57
Channel Copier options	57
Common Management Services options	60
Patch Manager options.	64
General options	65
Patch repository options	66
Patch options	68
Bulletin options	69
Patch group options.	70
Upgrade options	72
Patch Source options.	73
Patch Service options	73
Policy Manager options	74
Schedule formats	97
Proxy Administrator options	101
Proxy Server options.	114
Publisher options	119
Report Center options	123
Schema Manager options.	135
Transmitter options	141
Transmitter Administrator options	143
The admin options	144
The display options	145
The ldap options	146
The main options.	146
The publish options	148
The repeating options	149
The trans options	152

The users options	153
The workspace options	154
Tuner Administrator options	155
Using Tuner_ns commands from the command-line	160
Chapter 2 Tuner properties	163
Overview of tuner properties	164
List of tuner properties.	164
Chapter 3 Proxy properties	227
Overview of proxy properties	228
List of properties	229
Main proxy properties.	229
Server proxy properties	230
Cache proxy properties	232
Refresh proxy properties	233
Log proxy properties	233
Chapter 4 Transmitter properties	237
Overview of transmitter properties	238
List of properties	239
Transmitter extension properties	244
Chapter 5 Channel properties	247
Overview of channel properties	248
List of properties	248
Syntax for the schedule string.	259
Segment platform codes	260
Segment locale codes	261
Properties for the Policy Manager channel	266
Properties for Application Packager channels	262
Properties for the Tuner Update Manager channel.	266
Chapter 6 Parameters	269
Channel parameters	270
Schema Manager parameters	299

ADAM / AD LDS configuration parameters	299
Database configuration parameters	300
Chapter 7	
Channel states	303
Overview of channel states	304
States for Application Packager (any version) packages	304
States for Application Packager (version 4.6 or later) packages	305
Chapter 8	
Logging codes	307
Products without logging information	309
Log severity levels	309
Log ID ranges for specific channels	311
Action Request	312
Administration tools	313
Application Packager	314
Channel Copier	326
Common Management Services	328
Common Reboot Service	333
Content Replicator (Server Management)	334
Deployment Manager and Deployment Service (Server Management)	337
Infrastructure Service	363
Infrastructure Status Monitor	370
Inventory	371
Scanner Service	371
Inventory plug-in	374
Logging	377
Logging Service	378
Logging plug-in	379
Marimba Migration Module	379
Patch Management	381
Patch Manager	381
Patch Service	386
Policy Management	387
Policy Service	388
Policy Manager	392
Policy Service plug-in	394

Proxy	398
Proxy Administrator.	400
Report Center.	401
Schema Management	408
Setup and Deployment.	410
Storage.	415
Subnet Repeater Policy.	415
Transmitter.	416
Transmitter Administrator	425
Tuner	427
Tuner Administrator.	435
Tuner Packager	439
Chapter 9 Ports	441
List of ports.	442
Index	445

Preface

This reference provides information about Symphony Marimba Client Automation products in a single document. It assumes you are already familiar with the basics of using and administering these products, as described in the *Symphony Marimba Product Introduction* and the *Symphony Marimba Client Automation CMS and Tuner User Guide* (both are available on the Marimba Channel StoreMarimba Channel Store).

You should use this reference with the other documents and help for the various Symphony Marimba Client Automation components. This document covers information for the most current version of the product.

How to read this reference

The sections in this reference are not meant to be read sequentially. Typically, you use this reference to learn about a particular command-line option, tuner property, channel property, channel state, or logging code. Also, another document might have referred you to this reference. The following summary shows the sections (in order) in this reference.

- Chapter 1, “Command-line options,” describes the commands you can use for the various products and the various options that you can specify on the command line. You can use these commands and options at the Windows DOS prompt or in a UNIX shell (or in batch files or scripts).
- Chapter 2, “Tuner properties,” lists the properties that you can set for the tuner. You can set these properties when packaging a tuner or after the tuner is installed.
- Chapter 3, “Proxy properties,” lists the properties you can set for the proxy.

- Chapter 4, “Transmitter properties,” lists the properties you can set for the transmitter.
- Chapter 5, “Channel properties,” lists the properties that you can set for a channel. Each channel has properties that you can set when developing, packaging, publishing, or copying the channel.
- Chapter 6, “Parameters,” lists the parameters that you can set when you package an application using Application Packager — these parameters control how the packaged application is installed and launched. This section also lists the database and ADAM and AD LDS configuration parameters that you use when setting up an automated installation.
- Chapter 7, “Channel states,” describes the various channel states a channel can go through. These channel states correspond to the operations a channel performs and its success or failure.
- Chapter 8, “Logging codes,” lists the various logging codes that appear in the log files. Each tuner maintains log files for the tuner and the channels on it.
- Chapter 9, “Ports,” lists the port names that Symphony Marimba Client Automation products use and the default settings for those ports.

Related documentation

Symphony provides Symphony Marimba documents in PDF format. These documents are written for system administrators.

Product guides

This table lists guide descriptions for each product.

Guide	Description
<i>Symphony Marimba Client Automation Application Packager User Guide</i>	Provides information about packaging software for distribution to desktops or servers. This guide also includes information about command-line usage, policies, XML templates, and Windows system macros.
<i>Symphony Marimba Client Automation CMS and Tuner User Guide</i>	Provides information about the Common Management Services (CMS) and tuner infrastructure components. This guide also describes the tools and features you use to configure these components.

Guide	Description
<i>Symphony Marimba Client Automation Configuration Discovery Integration for CMDB Getting Started Guide</i>	Provides instructions about planning, installing, and configuring the Configuration Discovery integration. This guide also includes information about relationship classes and mappings, data exchanges, and reconciliation definitions.
<i>Symphony Marimba Client Automation Patch Management User Guide</i>	Helps you configure and administer Patch Management and the Patch Service plug-in. This guide also includes working with the patch repository, patches, patch groups, and custom patches, and deploying patches.
<i>Symphony Marimba Client Automation Policy Management User Guide</i>	Helps you configure and administer Policy Management and the Policy Service plug-in. This guide also includes integration procedures for directory services, such as Active Directory, ADAM / AD LDS and Oracle Directory.
<i>Symphony Marimba Client Automation Report Center User Guide</i>	Provides instructions about running queries of inventory information, configuring the Inventory and Logging plug-in, configuring endpoints, and integrating Report Center with other Symphony Marimba Client Automation applications.
<i>Symphony Marimba Client Automation Server Management User Guide</i>	Describes how to use Deployment Management and Content Replicator to control and monitor the distribution of content and applications across heterogeneous server platforms and data centers. Deployment Manager extensions to Report Center and Application Packager are also described.
<i>Symphony Marimba Client Automation Server Management CLI Reference Guide</i>	Provides syntax and usage information about the command-line options used with Content Replicator, Deployment Manager, and Application Packager. Using the SOAP interface feature is also described.
<i>Symphony Marimba Client Automation Transmitter and Proxy User Guide</i>	Provides information about the transmitters and proxy infrastructure components. This guide also describes the tools and features you use to configure these components.

General guides

This table lists descriptions of documentation that applies to all Symphony Marimba Client Automation products.

Guide	Description
<i>Database Schema documents</i>	Provides reference information about database schema, such as table names, field names, indexes, and primary, foreign, and unique key constraints.
<i>Definitive Software Library Administrator's Guide</i>	Provides a description of the Definitive Software Library and explains how the DSL is useful to you, how to use the DSL console, and how to access the DSL using Symphony Marimba Client Automation components, such as Report Center and Application Packager.
<i>Symphony Marimba Client Automation Device Management User Guide</i>	Describes how to use Symphony Marimba Client Automation to manage your mobile devices. This includes Scanner Service to perform inventory scans on your mobile device endpoints; Report Center to run queries against your scanned data; Application Packager, using the PDA Packager, to package and publish files and applications to mobile devices; and Policy Service to define subscription policies for your mobile devices.
<i>Symphony Marimba Client Automation Installation Guide</i>	Helps you design an infrastructure for your enterprise, which involves determining the machines to use for the various components and whether to purchase additional hardware and software. This book also instructions for a first-time installation of the product and its associated modules and how to upgrade from an earlier version.
<i>Marimba Concepts Guide</i>	Introduces you to Symphony Marimba Client Automation and defines basic concepts about core technology.
<i>Symphony Marimba Client Automation Reference Guide</i>	Provides reference information, such as command-line options, tuner properties, proxy properties, transmitter properties, channel properties, channel parameters, channel states, ports, and log IDs with associated log messages.

Using the documentation channel

If you are using a Windows platform, you can subscribe to the Symphony Marimba Client Automation Product Documentation channel. You can download the contents to a Windows computer from the Marimba Channel Store.

You can find the latest documentation channel in 9.0.00 - Current category of Channel Store.

Searching across the Symphony Marimba Client Automation documentation set

Symphony Marimba Client Automation comes with a large documentation set in PDF format, which can make it difficult to quickly find exactly what you need. With Adobe Reader version 7.0 and later, you can perform a full-text search across all of your PDFs that reside in the same directory, including subdirectories.

- **To search for a word or phrase in the PDFs contained in the Symphony Marimba Client Automation Product Documentation channel**
- 1 Ensure that you have installed the documentation channel as described in “Using the documentation channel” on page 14.
 - 2 If you do not have Adobe Reader version 7.0 or later, you can download the latest Adobe Reader version from www.adobe.com for free.
 - 3 Start Adobe Reader.
 - 4 Click the search icon (binoculars) in the toolbar.
If the search icon is missing, try right-clicking the toolbar to find more toolbar options. Different versions of Adobe Reader put the option in different places.
 - 5 After the Search window is displayed, select All PDF Documents in under **Where would you like to search?**
 - 6 Click the folder selection box and choose **Browse for Location**.
 - 7 Browse to the top-level folder that contains the PDF documents installed with the Symphony Marimba Client Automation Product Documentation channel.
 - 8 Type a search term in the **What word or phrase would you like to search for?** box and click **Search**.

The search tool searches for the term you entered in all of the PDFs in the chosen directory and its subdirectories and displays the results.

Using the Marimba Channel Store

The documentation is located on the Marimba Channel Store.

Locating the product documentation

You can find the latest product documentation on the Channel Store..

1 Command-line options

Each tuner comes with a `runchannel` program that you can run to start any channel on that tuner from the command line. The command-line options for Deployment Manager and Content Replicator are described in the *Symphony Marimba Reference Guide*, available on the Marimba Channel Store.

The following topics are provided:

- Overview (page 18)
 - The `runchannel` program (page 20)
 - The tuner program (page 25)
- Application Packager options (page 34)
- Certificate Manager options (page 54)
- Channel Copier options (page 57)
- Common Management Services options (page 60)
- Patch Manager options (page 64)
- Patch Source options (page 73)
- Patch Service options (page 73)
- Policy Manager options (page 74)
- Proxy Administrator options (page 101)
- Proxy Server options (page 114)
- Publisher options (page 119)
- Report Center options (page 123)

- Schema Manager options (page 135)
- Transmitter options (page 139)
- Transmitter Administrator options (page 143)
- Tuner Administrator options (page 155)
- Using Tuner_ns commands from the command-line (page 160)

Overview

Each tuner comes with a `runchannel` program that you can run to start any channel on that tuner from the command line. The following command starts the channel indicated by the URL (the channel for a Marimba component or a custom channel you created):

```
runchannel <channel_URL>
```

In the `runchannel` command, you can specify options related to the tuner or to the channel you are starting.

You can also start a channel and control the tuner in other ways by calling the `tuner` program from the command line. The following command starts the channel indicated by the URL:

```
tuner -start <channel_URL>
```

Like `runchannel`, `tuner -start` accepts options related to the channel you are starting. You can specify other tuner options (besides `-start`) in the `tuner` command.

Command syntax

In general, commands have the following syntax:

```
<command_name> [<command_arguments>] [<options>]
```

where the command name can (depending on the command) be followed by one or more arguments, such as a channel URL, and then by the command-line options described in this section. Each option begins with a hyphen (-) and can be followed by arguments; that is:

```
-<option_name> [<option_arguments>]
```

Some important notes regarding the syntax in this section follow:

- A plural term in the syntax stands for one or more occurrences of that item separated by spaces (unless a different separator character is noted in the description). Thus *<options>* stands for:

<option>1 <option>2 ... <option>N

- For easy reference, lists of options described in this section are ordered alphabetically (within sections that are ordered alphabetically by Marimba product name). However, an option of the form *-no<name>*, which negates the effect of the option *-<name>*, is grouped with its positive counterpart.
- For options that take multiple arguments, you must enclose the arguments in quotation marks, as noted in the descriptions of those options. You must enclose arguments that contain spaces in quotation marks.
- Command syntax and examples are sometimes shown on several lines (with lines beyond the first indented), but you must, of course, type each command from the command line without any break until the end of the command.

The following arguments are common to much of the syntax and have the form shown (unless specified otherwise in the option description).

<channel_URL>

has the form `http://<transmitter_name>[:<port>]/<channel_name>`. That is, the argument must be the full URL for the channel. Even though the channel URL (the URL you see when you hold the mouse pointer over a channel name) specifies the location of that channel on the transmitter, `runchannel` starts the channel from the tuner to which the channel was downloaded. If no port is specified, 5282 is assumed for the port on which the transmitter runs.

<directory> or <file>

is a full path to a directory or file. Argument names that end in *directory* or *file* also contain a full path to a directory or file. On a Windows system, these arguments include the drive letter; you cannot use server names in place of drive letters.

<host>

is a host name, such as `compl.myco.com`, or an IP address, such as `999.999.999.999`.

The runchannel program

The runchannel program lets you start a channel on the tuner from the command line. If the tuner isn't running, runchannel first starts the tuner. If you have more than one tuner installed, runchannel starts the tuner that's installed in the same directory as runchannel.

You normally use runchannel instead of the tuner program (that is, tuner -start <channel_URL>) to start a channel from the command line because:

- Unlike tuner -start, runchannel provides feedback from the channel. For example, the Publisher channel returns information about whether the publishing process worked.
- runchannel doesn't display the graphical user interface of the channel you are starting, and by default it also doesn't display the user interface of the tuner. On the other hand, tuner -start displays the channel's interface, and it displays the tuner's interface by default.

Thus runchannel is better suited for a command-line session in which you repeatedly interact with a channel (where the commands build on each other) and for running a channel from a batch file or a script. If you do want the channel's interface displayed, or you just want to set tuner options without starting a channel, use tuner instead of runchannel.

Note: You should avoid launching multiple instances of runchannel because the system can run out of resources.

The runchannel and tuner programs are located in a directory in the tuner's installation directory. As always, when you are issuing commands from the command line, switch to the directory in which the executable file is located, or set that directory in the path your system is configured to use for finding commands.

The syntax for using runchannel follows:

runchannel [<tuner_options>] <channel_URL> [<options>]

or

runchannel [@<file>]

The tuner options that you can specify before the channel URL are described in the next section. (In some cases, options have an equivalent in the `tuner` command.) You can specify the `@<file>` option for any channel, but all options must be in the specified file. You cannot specify other options on the command line.

Command-line options

This section lists the command-line options for the `runchannel` program.

`-help`

lists and briefly describes all the tuner options that work with `runchannel`.

`-p`

starts the tuner's primary channel. If you use this option, you don't need to specify the channel URL.

`-quit`

quits the tuner. (You don't need to specify a channel URL in the `runchannel` command in order for this option to work.) If the tuner is running as a service on Windows, the service stops, but it restarts when you restart the computer.

`-rpc [sslonly | ssl | nossl] [[<host>:]<port>]`

turns RPC server support on. Tuners must allow RPC connections to be accessed remotely. For example, your tuner must use RPC if you want to run a transmitter or Publisher.

The kind of RPC connections to accept are specified by `sslonly | ssl | nossl`: only SSL (`sslonly`), both SSL and unencrypted connections (`ssl`), or only unencrypted connections, which is the default (`nossal`).

The host and port number to use are specified by `[<host>:]<port>`. Specifying the host is useful only for machines with multiple network interfaces. The default host is `localhost`.

You can specify the port number as `0` if you want a free port to be allocated automatically. The default port number is `7717`. If you allow both SSL and unencrypted RPC connections, the connections share the same port (`7717` by default).

`-subscribe`

`-nosubscribe`

control what happens if the channel is not already in the tuner’s workspace directory: whether to subscribe to the channel (in which case dialog boxes can be shown) or to fail rather than subscribe. `-nosubscribe` is the default. If the channel comes from a transmitter that requires a subscription user name and password, you can use `-user <user_name>` and `-password <password>` after `-subscribe`.

`-timeout <number_of_seconds>`

specifies the amount of time (in seconds) to wait for a tuner response before timing out. The default timeout is 60 seconds. The `runchannel` command tries to send its request to the tuner every 15 seconds until it reaches the timeout. For example, if the timeout is set to 60 seconds, `runchannel` tries 4 times ($60/15=4$) before timing out. When you use the `-timeout` command line option with `runchannel` command, the `-timeout` option works only when it is used in conjunction with other `runchannel` operations. For example, the `runchannel -timeout 100 [channelURL]` implies that `runchannel` waits for 100 msecs to connect to the tuner to start the channelURL.

`-tuner <tuner_option> | "<tuner_options>"`

passes one or more options to the tuner, if the tuner has to be started; these options can be any of the `tuner` command options described in “Tuner options” on page 27. The `-tuner` option is useful for specifying those tuner options that you can’t enter directly (before the channel URL) in the `runchannel` command.

`-update`
`-noupdate`

control whether to update a channel before starting it. This option applies to the primary channel and to other channels that you are starting.
`-noupdate` is the default.

`-ws <directory_name>`

specifies the name of the workspace directory for the tuner to use. For more information, see the option with the same name in “Tuner options” on page 27.

Options specific to the channel you are starting can follow the channel URL in the `runchannel` command. The options that you can specify depend on the channel you are starting. This section describe the options available for various Marimba products. Depending on the product, some of these options may be required.

For example:

```
runchannel http://mytrans/mychannel
```

starts a custom channel having the specified URL (on the mytrans transmitter). Before you can start and run the channel, you must first subscribe to it using the tuner.

```
runchannel http://mytrans/publisher -d mychanneldirectory  
http://mytrans/MyChannel
```

starts the Publisher channel and publishes the contents of mychanneldirectory. This command assumes that you have Publisher installed in the tuner and that you have a custom channel in the mychanneldirectory that you are ready to publish.

```
runchannel -ws one -rpc 1234 http://mytrans/mychannel  
-someoption
```

starts the tuner using the RPC port 1234. This command also specifies a workspace directory named one. The tuner starts the custom channel having the specified URL and sends it the -someoption option.

Return codes

If you want to know if a `runchannel` command succeeded, you can monitor the `runchannel` return or exit code. The return code is written to the `launch.log` file along with the log entries. (By default, the `launch.log` file is created in the tuner workspace. For any value other than “no,” the `launch.log` file is created. However, if you specify the value as “no,” the file is not created.) For more information about log entries, see “Logging codes” on page 307. The return codes are 0, for success, and -1 through -12, for failure, as described in the following table:

Table 1-1: Return Codes

Code	Description
0	<ul style="list-style-type: none"> ▀ <code>runchannel -help</code> was requested (and provided). ▀ A channel was started and completed an operation successfully.
-1	<ul style="list-style-type: none"> ▀ In the tuner’s installation directory, the <code>properties.txt</code> file was missing from the <code>lib/tuner/</code> directory. ▀ There was an incorrect <code>runchannel</code> command-line argument. ▀ An exception occurred while trying to start the tuner. ▀ There were notifications related to channel exceptions, RPC disconnects, or RPC timeouts.

Table 1-1: Return Codes (Continued)

Code	Description
-2	An unexpected exception was caught. An associated stack trace gives more details. If you see this code while publishing a signed channel, then make sure that the cert DN (certificate's distinguished name) is specified correctly.
-3	Command was canceled.
-4	I/O error. Log entries should give more details, but this error can mean that the command failed to get data channel properties (that is, a communication problem between the endpoint and transmitter exists). This error can also mean that the command failed to move internal directories in or out during a publish or install operation
-5	Publish error. A publish or unpublish command failed in the publish process. The actual error is described in the logs.
-6	Cleanup error. Not currently used.
-7	Update error. An install or uninstall command failed in the installation process. The actual error is described in the logs.
-8	Concurrency error. This error can occur only when the application is used as a library. The same instance of a kernel was used to perform multiple publish or installation commands simultaneously.
-9	Exec error. An error occurred while trying to execute a pre- or post-installation command.
-10	Unsigned error. Not currently used. Attempting to install a channel that is not properly signed returns a -3 (canceled) code, but a log entry indicating that the channel was not signed correctly also exists.
-11	Unknown error. This error should not be returned. If this error is returned, the logs contain entries indicating the problem.
-12	No-such-channel error. An error occurred while getting the data channel list from a transmitter. When installing, the data channel is what the application publishes to the transmitter during a publish command.

The tuner program

You can run the tuner from the command line with the tuner program. The syntax for using the tuner program follows:

```
tuner [<tuner_options>]
```

or

```
tuner [@<file>]
```

The tuner options you can specify can are described in “Tuner options” on page 27. In `[@<file>]`, *file* is the name of the file that contains the options to use when starting the tuner. All options must be in the specified file. You cannot specify other options on the command line.

Note: On Windows systems, you can also set the command-line options in the properties for a shortcut (such as a desktop icon).

Tuner options

This section lists the options for the tuner program.

`-admin "<user_name>,plain:<password>"`

defines an administrator user name and password and sets the `marimba.tuner.admin` property in the `prefs.txt` file. The next time you restart the tuner, the password is encrypted and rewritten in the file. You can use this option if you plan to administer the tuner using Tuner Administrator. You use this option when starting the tuner.

`-anonymize`

if you use a Ghost image to deploy tuners, removes certain properties, files, and registry entries to avoid having multiple endpoints report that they have the same machine ID and tuner ID. For more information, see the appendix about using a Ghost image to deploy products in the *Marimba Guide*, available on the Marimba Channel Store. If the tuner is run in `-anonymize` mode, it sets the tuner workspace ID as NULL, so that the tuner is started with a different workspace ID if the tuner is imaged and installed on a different host.

You must run the `-anonymize` command only on a tuner whose password encryption feature has never been enabled. Otherwise, when the tuner is imaged, encrypted passwords cannot be decrypted on the host once the image is installed.

Note: The `-anonymize` option works only on Windows platforms.

You must set the `marimba.security.token.enable` property to false before you can use the `-anonymize` option. This property is set in the `prefs.txt` file to disable the password encryption feature of the tuner.

A common approach is to include the tuner in the golden image that gets deployed when a new machine is provisioned. The `-anonymize` option allows users to retain that approach. Using the `-anonymize` option ensures that the workspace ID becomes NULL and gets regenerated when the tuner comes up, thus ensuring that every new machine gets a new ID.

`-certPassword <password>`

specifies the password to use with the server certificate for SSL. Using this option is an alternative to setting the `marimba.tuner.rpc.certpw` property with the base64-encoded SSL password. You use this option when starting the tuner.

`-console`

opens a console window at tuner startup. This window displays internal tuner messages. If you use this option, it should be the first option you specify. This option is available for Windows systems only.

`-debugOnConsole`

prints debug output for the tuner program on `stdout` of the system console where the tuner was started. This option is available for UNIX systems only.

Example: `./tuner.sh -debugOnConsole`

`-display`

`-nodisplay`

control whether to run the tuner with a graphical user interface. You use this option when starting the tuner.

`-installchannel <carfile1_path> [url <channel1_URL>] [<carfile2_path> [url <channel2_URL>] <carfile3_path> [url <channel3_URL>] ...]`

installs one or more channel archive (CAR) files into the tuner's workspace. Each CAR file represents a channel. Optionally, you can specify a URL for each channel.

<carfile1_path> is the full path to the CAR file that you want to install, and <channel1_URL> is the update URL for the channel.

The CAR files are created using Channel Copier or Transmitter Administrator by copying channels from a transmitter to CAR files. If you are creating a CAR file by copying channels from a package or publish directory (not from a transmitter), no URL is associated with the channel. Specify a channel URL using the command-line option `url <channel1_URL>`.

The CAR file is installed only if no channel with the same URL as one in the CAR file exists in the tuner's workspace, or if the channel exists but is in the "available" state. You can use `-installchannel` both when starting up a tuner and by sending it to a tuner that is already running.

Limitation: When installing a channel with this command, a duplicate desktop shortcut associated with the channel is created if Channel Manager is currently running. This limitation applies only if a desktop shortcut was specified for this channel when it was published. This results from Channel Manager and the tuner both attempting to create a desktop shortcut upon installing the channel.

Note: On UNIX platforms, you must place the Java command-line options (`-java`, `-jit`, `-nojit`) ahead of any other command-line options. For `runchannel`, you must observe this restriction within any value given for the `-tuner` command-line option.

`-intro [<channel1_URL>]`
`-nointro`

control whether an intro channel is started the first time the tuner is started. `-intro` runs the specified channel as the intro channel; if no channel is specified, it runs the channel identified by the `marimba.intro.url` tuner property. You use this option when starting the tuner.

`-java <java_arg> | "<java_args>"`

sends one or more arguments to the Java virtual machine. When specifying several arguments, enclose all of arguments in quotation marks.

`-jit`
`-nojit`

control whether to turn on the JIT compiler. `-jit` is the default.

```
-keeprunning
```

specifies that the tuner keeps running even after all channels stop. You can set the default using the `marimba.tuner.keeprunning` tuner property. If this property is not set, the default behavior depends on the platform: on Windows, tuners keep running; on Solaris, tuners quit. You use this option when starting the tuner.

```
-locale <language>_<country>
```

starts the tuner with a new locale and sets the `user.locale` property accordingly. All channels in the tuner's workspace directory are set to be updated when run again. You use this option when starting the tuner.

`<language>` is the two-letter ISO 639 language code (always lowercase).

`<country>` is the two-letter ISO 3166 country code (always uppercase).

Sample values are `en_US`, `fr_FR`, and `es_ES`.

Note: It is not recommended to set this property in the Launch Arguments of a profile in the Profile tab.

```
-logo
```

```
-nologo
```

control whether to display the logo window at tuner startup. `-logo` is the default. You use this option when starting the tuner.

```
-minimal
```

starts the tuner with a minimal configuration, which minimizes the RAM size of the tuner. This option has the same effect as specifying `-noprimary` and `-nologo`.

```
-multicast [[<host>:]<port>]
```

```
-nomulticast
```

control whether to turn on the tuner's multicast feature, which channels use to discover available tuners and transmitters on the network.

Optionally, you can specify the multicast address and port on which to communicate with other systems. The default address is

`228.200.200.200:3343`. You use this option when starting the tuner.

```
-nowin
```

starts the tuner with a minimal configuration. This option is the same as `-minimal` and is provided for backward compatibility with version 2.0 shortcuts.

-primary <channel_URL>
-noprimary

control whether to start the primary channel at tuner startup. -primary starts the specified channel as the primary channel. If these options are not used, the default behavior is to start the primary channel specified by the marimba.primary.url tuner property (the Channel Manager by default). You use this option when starting the tuner.

-quit

quits the tuner. If the tuner is running as a service on Windows, the service stops, but it restarts when you restart the computer.

-redirect [<file>]

saves output from the tuner to the specified file. If no file is specified, the tuner saves its output to program.out.

-rpc [sslonly | ssl | nossl] [[<host>:]<port>]

-norpc

turn RPC server support on or off. Tuners must allow RPC connections to be accessed remotely. For example, your tuner must use RPC if you want to run a transmitter or Publisher. You use this option when starting the tuner.

The kind of RPC connections to accept are specified by sslonly | ssl | nossl: only SSL (sslonly), both SSL and unencrypted connections (ssl), or only unencrypted connections, which is the default (nossl).

The host and port number to use are specified by [<host>:]<port>. Specifying the host is useful only for machines with multiple network interfaces. The default host is localhost.

You can specify the port number as 0 if you want a free port to be allocated automatically. The default port number is 7717. If you allow both SSL and unencrypted RPC connections, the connections share the same port (7717 by default).

-security
-nosecurity

control whether to enforce the default tuner security policies. -security is the default. With security off, you are not able make SSL connections, and signed channels can have problems running because they are not able to use certain features (such as file system access or printing). You use this option when starting the tuner.

-show <channel_URL>

displays the specified URL in the Channel Manager Browse page and follows that URL. This option works only if the tuner is already running.

`-ssljava` (deprecated for 6.0.3)

forces the tuner to use Java classes for SSL. The default is to use the native libraries if they are available. You use this option when starting the tuner.

`-start <channel_URL> [<options>]`

starts the specified channel. If the channel is not already subscribed, this option subscribes and starts the channel. Following the URL, you can specify command-line options to send to the channel. If you specify more than one tuner option in the tuner command, `-start` must be the last option you specify because all subsequent options on the command line are sent to the channel.

For reasons stated in “The runchannel program” on page 22, you generally use `runchannel` instead of `tuner -start` to run channels from the command line.

The options that you can specify for various channels corresponding to Marimba products are described in this section. However, because of the emphasis on using `runchannel`, `tuner -start` is not shown as a way of starting the channel—and even though `tuner`

`-start` can work, some options may not work with it. (The only options that work with `tuner -start` are those that are indicated as such.)

`-testCerts`

enables test certificates. This option allows the tuner to accept installed test certificates rather than real certificates.

`-trust <subject>|<issuer>`

means any channel using a certificate that matches `<subject>|<issuer>` is trusted by the tuner. Either the `<subject>` or `<issuer>` can be the string "any", which allows the tuner to trust any subject or issuer.

For example:

- "cn=Acme, Inc."|"ou=VeriSign Trust Network" refers to a specific certificate.
- "cn=Acme, Inc."|"any" refers to any Acme certificate.
- "any"|"ou=VeriSign Trust Network" refers to any certificate issued by VeriSign.
- "any"|"any" means any channel with a certificate is trusted.

You use this option when starting the tuner.

-update
-updatefrom

changes the URL from which a channel gets the updates. Using this option triggers an update of the channel after the URL is changed.

For example: Tuner.exe -updatefrom -current [existing channel URL] -new [new channel URL]

-noupdate

control whether to update a channel before starting it. This options applies to the primary channel and to other channels that you are starting.

-noupdate is the default. You use this option when starting a channel.

-v

turns on the tuner's verbose mode, which provides useful information regarding speed, garbage collection, class loading, and so on.

-ws <directory_name>

specifies the name of a workspace directory the tuner uses. This options is helpful when you want to run multiple tuners that use different workspace directories. For example, you can have the default tuner that uses the workspace directory named after the keyword and a custom tuner that uses a workspace directory named `mycompany`. The workspace directory is in the `.marimba` directory. You use this option when starting the tuner.

Once you run a tuner using the `-ws` option, you must always specify the `-ws` option when communicating with that tuner.

Running the tuner as a service on Windows

You can run the tuner as a service on Windows if you have channels you want to run continuously. For example, you can have a transmitter that must be running all the time. The advantage to running the tuner as a service on Windows is that you can configure the tuner (and its primary channel) to start automatically when the computer is started. You can also configure the tuner not to run (or to be started manually) when the computer is started.

During installation, an option configures the tuner to run as a service. You can also install the tuner as a service from the command line.

With fresh installations of the tuner, the tuner runs as a service with the default Startup Type of Automatic. But if the tuner is installed or reinstalled from the command line to run as a service and if the startup type is not mentioned explicitly while executing the command, then the default Startup Type will be set to Manual and the tuner will not start automatically after system startup unless it is started manually.

Note: Installing and running the tuner as a service is supported only on the system account. Make sure you are logged on to the system with administrator rights when you install a tuner.

If you do not specify the attended option when this command is run, you cannot interact with the tuner through the desktop.

Note: By default, the tuners that are installed as services using a profile or a unified installer are configured to allow user interaction through the desktop.

► To install the tuner so it runs as a service (from the command line)

- 1 Open a command prompt and go to the directory where you placed the tuner installer.
- 2 Type the following and press Enter.

```
tuner -service install {automatic | disabled | manual} {attended | unattended} {prefs | props} <service_arguments>
```

where:

automatic

configures the tuner to automatically start during system startup.

disabled

prevents the tuner from starting unless it is specifically started using the Services utility in the Control Panel.

manual

allows users to start (or stop) the tuner from the command line, Start menu, or Control Panel.

attended

allows users to interact with the tuner through the desktop.

Note: If you do not explicitly specify this option while executing the `tuner -service install` command, then you cannot interact with the tuner through the desktop.

unattended

allows the tuner to perform operations when no users are logged on to the system.

`prefs`

stores any service arguments in the `prefs.txt` file.

`props`

stores any service arguments in the `properties.txt` file.

`service_arguments`

can be any number of tuner command-line arguments. For descriptions of command-line arguments, see “Tuner options” on page 27. These arguments are stored in the `marimba.launch.NTServiceArgs` tuner property.

For example:

```
tuner -service install automatic attended prefs -nologo
```

installs the tuner as a service that automatically starts during system startup, allows desktop interaction with users, and does not display the tuner logo screen. The argument `-nologo` is in the `prefs.txt` file.

► To uninstall the tuner running as a service

- From the command line, type the following and press Enter:

```
tuner -service uninstall
```

This command uninstalls the tuner and removes it from Service Control Manager (SCM) registration.

By default, if an explicit option 'attended' is not specified when this command is run, users will not be able to interact with the tuner through the desktop. Note: Tuners that are installed as services using profile or unified installer are configured to allow user interaction through the desktop by default.

Channel operation options

This section lists the channel operation options. Channel operation options can be passed to the tuner at any time, and require the URL of the channel on which to perform the operation.

-remove <channel_URL>

removes the specified channel from the tuner's workspace.

-stop <channel_URL>

stops the specified channel.

-subscribe <channel_URL>

subscribes the tuner to the specified channel.

-unsubscribe <channel_URL>

unsubscribes the tuner from the specified channel.

update <channel_URL>

updates the specified channel. This option has no dash (-) preceding it;

-update controls whether to update a channel before starting it.

Application Packager options

You can use the command-line options for the following Application Packager components:

- “Custom Application Packager command-line options” on page 37
 - “File Packager command-line options” on page 38
 - “Packager for Shrinkwrap Windows Applications command-line options” on page 40
 - “Windows Installer Packager command-line options” on page 42
 - “.NET Packager command-line options” on page 43
 - “PDA Packager command-line options” on page 47
- In addition, you can use the command line to do the following:
- “Upgrading channels” on page 52
 - “Checking the version of Application Packager and channels” on page 53
 - “Applying an XML template file to channels” on page 53

- “Starting packaged applications from the command line” on page 54
- “Staging MSI applications on endpoints” on page 56

Custom Application Packager command-line options

You can start Application Packager with the `runchannel` program and specify Custom Application Packager options as follows:

```
runchannel <App_Packager_URL> -custompackage <channel_name>  
<channel_directory> [-defaults <XML_template_file_path>]
```

creates a new channel with the specified name in the specified directory.

`<channel_name>` specifies the name that you want to give the channel.

`<channel_directory>` specifies the full path of the directory where you want to place the contents of the new channel that you are creating.

`[-defaults <XML_template_file_path>]` specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Example (Windows)

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-custompackage MyCustomChannel C:\Channels\MyCustomChannel  
-defaults C:\Channels\Templates\custom.xml
```

Example (UNIX)

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-custompackage MyCustomChannel /opt/source/pkgs/MyCustomChannel  
-defaults /opt/source/pkgs/templates/custom.xml
```

File Packager command-line options

You can start Application Packager with the `runchannel` program and specify File Packager options as follows:

```
runchannel <App_Packager_URL> -filepackage <options>
```

where the options can be any of those described in this section.

Packaging a set of directories and files

```
runchannel <App_Packager_URL> -filepackage <title>
<channel_directory>
<directory_list> [-defaults <XML_template_file_path>] [-props
<property_list>] [-publish <tx location>] [-txauth
<username>:<password>]
```

creates a channel in the specified directory from the source directories listed in *<directory_list>*.

<title> specifies the title of the package

<channel_directory> specifies the full path of the directory where you want to place the contents of the new channel you are creating.

Each directory in *<directory_list>* has the form

```
<directory> [as <install_directory>]
[prompt "<text>"]
[ref]
[ntperm]
```

Either specify an install directory or prompt the user for a directory where the files should be installed.

<directory> is the full path of the directory containing content to package.

as *<install_directory>* specifies the directory on the user's system where the packaged files are installed when the channel is run.

prompt "*<text>*" queries the user for the directory in which to install the packaged files when the channel is run, displaying the specified text as a prompt for the user's input. The text should be a message asking the user to specify the target directory on the user's system.

ref packages by reference the specified directory.

Default value: false

ntperm ntperm specifies whether to capture Windows file permissions when packaging the specified directory. This option is available only in Windows NT, Windows 2000, and Windows XP. Windows supports two kinds of file systems: FAT and NTFS. The latter offers more advanced file permission settings. To take full advantage of this option, both the person who is creating the file package and the user must be using an NTFS file system. Otherwise, only basic Windows file permissions and attributes (read-only, archive, hidden, system) are preserved.

Default value: false

Note: Users installing the channel must have administrative privileges to set permissions.

`[-defaults <XML_template_file_path>]` specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used.

`[-props <property_list>]` specifies the list of properties to be set for the package. `<property_list>` specifies the list of properties to be set and it takes the format

`<property1>=<value1>:<property2>=<value2>:<property3>=<value3>`

`[-publish <tx_location>]` specifies whether you want to publish the packaged channel. If your transmitter is configured with publish credentials, use `-txAuth` option to specify the publish credentials

Example (UNIX)

```
runchannel http://trans.acme.com:5282/ApplicationPackager -filepackage  
MyFilePackage C:\Channels\MyFilePackage C:\ContentFiles\src1 as C:\Install\src1  
prompt "Where to install src1?" ref ntperm C:\ContentFiles\src2 as C:\Install\src2  
prompt "Where to install src2?" ref ntperm -  
props=testprop1=testvalue1:testprop2=testvalue2 -publish http://myserver:5282/filepackage1 -txAuth user1:password
```

Repackaging the contents of an existing File Packager channel

```
runchannel <App_Packager_URL> -filepackage -repackage [-publish]  
<channel_directory> [-defaults <XML_template_file_path>]
```

repackages the contents of an existing File Packager channel in the specified directory. Optionally, you can immediately publish the repackaged channel by using the `-publish` option.

`[-defaults <XML_template_file_path>]` specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Example (Windows)

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-filepackage -repackage C:\Channels\MyFilePackage
```

Example (UNIX)

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-filepackage -repackage /home/user1/MyFilePackage
```

Packager for Shrinkwrap Windows Applications command-line options

You can start Application Packager with the `runchannel` program and specify Packager for Shrinkwrap Windows Applications options as follows:

```
runchannel <App_Packager_URL> -shrinkwrappackage <options>
```

where the options can be any of those described in this section.

Creating pre-install and post-install snapshots

First, you take two snapshots of the system: one before and one after installing the application that you want to package. Later, Packager for Shrinkwrap Windows Applications compares the two snapshots to determine the files, registry entries, and other items that make up the packaged application or channel.

Usually, this is a three-step process:

- 1 Create the pre-install snapshot.
- 2 Install the application.
- 3 Create the post-install snapshot.

You can create a snapshot as follows:

```
runchannel <App_Packager_URL> -snapshot <snapshot_file_path>  
<XML_template_file_path>
```

creates a snapshot of the system using the filters in the specified XML file and saves the snapshot to the specified file path.

`<file_path>` specifies the full path and name for the file where you want to save the snapshot.

`<XML_template_file_path>` specifies the XML template file that contains the filters you want to use for the snapshot. For more information, see the *Application Packager User Guide*.

For example:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-snapshot C:\SystemSnapshots\preinstall.msp  
C:\SystemSnapshots\Templates\snapshot.xml
```

Comparing the snapshots and creating a channel

After taking pre-install and post-install snapshots, you compare the snapshots and create the channel using the following command-line options:

```
runchannel <App_Packager_URL> -shrinkwrappackage <channel_name>  
<presnapshot_file_path> <postsnapshot_file_path>  
<channel_directory> [-includedeletes <args>] [-defaults  
<XML_template_file_path>]
```

creates a new channel with the specified name in the specified directory.

<channel_name> specifies the name that you want to give the channel.

<presnapshot_file_path> specifies the full path of the pre-snapshot file that you previously created.

<postsnapshot_file_path> specifies the full path of the post-snapshot file that you previously created.

<channel_directory> specifies the full path of the directory where you want to place the contents of the new channel that you are creating.

[-includedeletes <args>] specifies whether or not to capture the deletion of items and to include them in the channel. <args> specifies one or more of the following keywords:

- filesystem
- registry
- metabase

If you omit this option, the packager does not capture the deletion of items.

[-defaults <XML_template_file_path>] specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Example:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-shrinkwrappackage MyShrinkwrapChannel  
C:\SystemSnapshots\preinstall.msp  
C:\SystemSnapshots\postinstall.msp  
C:\Channels\MyShrinkwrapChannel -includedeletes filesystem  
registry -defaults C:\Channels\Templates\shrinkwrap.xml
```

Windows Installer Packager command-line options

You can start Application Packager with the `runchannel` program and specify Windows Installer Packager options as follows:

```
runchannel <App_Packager_URL> -wipackage <MSI_database_path>  
<channel_directory> [-include <true | false>]  
[-byreference <true | false>] [-defaults <XML_template_file_path>]
```

`creates a new channel with the specified name in the specified directory.`

`<MSI_database_path>` specifies the full path of the MSI database file (with the `.msi` extension) that you want to package.

`<channel_directory>` specifies the full path of the directory where you want to place the contents of the new channel that you are creating.

`[-include]` specifies whether or not you want to include in the channel all the files that are in the same directory as the MSI file. If you omit this option, the other files in that directory are not included in the channel.

`[-byreference]` specifies whether or not you want to package files by reference.

`[-defaults <XML_template_file_path>]` specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Example:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-wipackage C:\MSI\W2Kapp\application.msi  
C:\Channels\MyWindowsInstallerChannel -defaults  
C:\Channels\Templates\msi.xml
```

Repackaging the contents of an existing Windows Installer Packager channel

```
runchannel <App_Packager_URL> -wipackage <channel_directory>  
-repackage [-include <true | false>] [-byreference <true | false>]  
[-defaults <XML_template_file_path>]
```

repackages the contents of an existing Windows Installer Packager channel in the specified directory.

[-include] specifies whether or not you want to include in the channel all the files that are in the same directory as the MSI file. If you omit this option, the other files in that directory are not included in the channel.

[-byreference] specifies whether or not you want to package files by reference.

[-defaults <XML_template_file_path>] specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Before repackaging, save the previous configuration of the Windows Installer package in an XML template file so that it can be used on a subsequent repackage. After the Windows Installer package has been repackaged with updated content, the XML template file is applied to restore the previous configuration. See the instructions for saving the Windows Installer package configuration in the *Application Packager User Guide*.

.NET Packager command-line options

You can start Application Packager with the runchannel program and specify .NET Packager options as follows:

```
runchannel <App_Packager_URL> -dotnetpackage  
-packagedir <channel_directory>  
[-channelname <channel_name>]  
[-ini_file_objects <true | false>]  
[-assembly_policy_config_objects <true | false>]  
[-sourcedir <source_directory_path>  
[as <target_directory_path>]  
[prompt <install_directory_prompt_text>]  
[filebyref <true | false>]  
[ntfileperm <true | false>]
```

```
]  
[-defaults <XML_template_file_path>]
```

creates a new channel with the specified name in the specified directory.

<channel_directory> specifies the full path of the directory where you want to place the contents of the new channel that you are creating.

<channel_name> specifies a name for the channel.

-ini_file_objects specifies whether or not INI files should be treated as special file objects. If this option is set to true, Application Packager parses the INI files and captures any changes made to the key and value pairs in them. Any changes found by Application Packager are merged with the existing file found at the endpoint. If you want Application Packager to process INI files as ordinary files and not merge changes, you should set this option to false. For more information, see the *Application Packager User Guide*.

Default value: true

-assembly_policy_config_objects specifies whether or not you want assembly policy configuration files to be treated as special file objects. Application Packager parses the configuration files and captures any changes made to them. Any changes found by Application Packager are merged with the existing file found at the endpoint. If you want Application Packager to process configuration files as ordinary files and not merge changes, you should set this option to false.

Default value: true

[-defaults <XML_template_file_path>] specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

For each source directory that you want to include in the channel, you specify the following information:

<source_directory_path> is the full path of the directory containing content to package.

as <target_directory_path> specifies the directory on the endpoint where the packaged files are installed when the channel is run.

`prompt "<install_directory_prompt_text>"` queries the endpoint user for the directory in which to install the packaged files when the channel is run, displaying the specified text as a prompt for the user's input. The text should be a message asking the user to specify the target directory on the user's system.

`filebyref` specifies whether or not to package files (in the specified directory) by reference. For more information about packaging files by reference, see the *Application Packager User Guide*.

Default value: `false`

`ntfileperm` specifies whether or not to capture Windows file permissions when packaging the specified directory. This option is available only on Windows NT, Windows 2000, and Windows XP. Windows supports two kinds of file systems: FAT and NTFS. The latter offers more advanced file permission settings. To take full advantage of this option, both the person who is creating the file package and the user must be using an NTFS file system. Otherwise, only basic Windows file permissions and attributes (read-only, archive, hidden, system) are preserved.

Default value: `false`

Note: Users installing the channel must have administrative privileges to set permissions.

Example:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-dotnetpackage -packagedir C:\Channels\MyDotNetPackage  
-channelname My_Dot_Net_Pkg -sourcedir C:\AssemblyFiles\src1 as  
C:\Install\src1 prompt "Where to install src1?" filebyref true  
ntfileperm true -sourcedir C:\AssemblyFiles\src2 as  
C:\Install\src2 prompt "Where to install src2?" filebyref true  
ntfileperm true
```

Repackaging the contents of an existing .NET Packager channel

```
runchannel <App_Packager_URL> -dotnetpackage -packagedir  
<channel_directory>  
[-sourcedir <source_directory_path>
```

```
[as <target_directory_path>]
[prompt <install_directory_prompt_text>]
[filebyref <true | false>]
[ntfileperm <true | false>]
]
-repackage
[-defaults <XML_template_file_path>]

repackages the contents of an existing .NET Packager channel in the
specified directory.
```

For each source directory that you want to include in the channel, you specify the following information:

<source_directory_path> is the full path of the directory containing content to package.

as <target_directory_path> specifies the directory on the endpoint where the packaged files is installed when the channel is run.

prompt "<install_directory_prompt_text>" queries the endpoint user for the directory in which to install the packaged files when the channel is run, displaying the specified text as a prompt for the user's input. The text should be a message asking the user to specify the target directory on the user's system.

filebyref specifies whether or not to package files (in the specified directory) by reference. For more information about packaging files by reference, see the *Application Packager User Guide*.

Default value: false

ntfileperm specifies whether or not to capture Windows file permissions when packaging the specified directory. This option is available only on Windows NT, Windows 2000, and Windows XP. Windows supports two kinds of file systems: FAT and NTFS. The latter offers more advanced file permission settings. To take full advantage of this option, both the person who is creating the file package and the user must be using an NTFS file system. Otherwise, only basic Window file permissions and attributes (read-only, archive, hidden, system) are preserved.

Default value: false

Note: Users installing the channel must have administrative privileges to set permissions.

[`-defaults <XML_template_file_path>`] specifies the XML template file that you want to use when creating the new channel. If you omit this option, the XML template file set as the default for Application Packager is used. For more information, see the *Application Packager User Guide*.

Before repackaging, save the previous configuration of the Windows Installer package in an XML template file so that it can be used on a subsequent repackage. After the Windows Installer package has been repackaged with updated content, the XML template file is applied to restore the previous configuration. See the instructions for saving the .NET Packager Channel configuration in the *Application Packager User Guide*.

PDA Packager command-line options

This section describes how to use the command line to create packages for your mobile devices. You can use the `-pdapackage` command-line option as an alternative to using the Application Packager PDA Packager graphical user interface.

Syntax for runchannel. Use the following syntax for the `runchannel` program:

```
runchannel <App_Packager_URL> -pdapackage <arg1> <arg2>
```

where `<App_Packager_URL>` is the URL of the Application Packager channel that is subscribed to by the local tuner. It has the form:

```
http://<transmitter_name>[:<port>]/ApplicationPackager
```

That is, it must be the full URL for the channel. Even though the channel URL specifies the original location of that channel on the transmitter, `runchannel` still starts the channel from the tuner to which it's been downloaded.

Syntax for the `-pdapackage` option. The arguments for the `-pdapackage` option are described in the rest of this section. The following options and arguments apply to both packaging content and packaging applications:

```
-pdapackage -packagedir <channel_directory>  
[-channelname <name_of_channel>] [-repackage]
```

specifies that you want to create a package for a mobile device. The `-packagedir` option creates a channel in the directory specified by `<channel_directory>`. This directory name is also used as the URL name when the package is copied to your transmitter (unless you use Channel Copier to change the URL name). Regardless of whether you want to package an application (CAB file) or content (files in a directory), specify both the `-pdapackage` and `-packagedir` options.

If you are packaging the channel for the first time, be sure to use the option `-channelname` to specify the name that you want to give the channel.

If you are repackaging the channel, rather than packaging the channel for the first time, you can omit the `-channelname` option. In this case, be sure to use the `-repackage` option.

Then, if you are creating a package for the first time, depending on the type of package you want to create, use one of the following groups of command-line options (if you are repackaging, use the following options only if you want to add directories or CAB files or change another setting):

- **Content files.** For packaging content (one or more directories of files), use the following options:

`-sourcedir <source_directory>` specifies the full path to the directory that contains the content to be packaged.

`as <target_directory>` specifies the full path to the directory on the user's system where the packaged files are installed when the channel is run. In most cases, you can use a short path. For example, if you package `C:\Program Files\Microsoft ActiveSync\MyApplication`, you probably want the install path to be `\MyApplication`.

`[filebyref {true|false}]` means, if set to true, that the source directory is packaged by reference in the specified directory. Use this argument if you want to save disk space on the packaging machine.

Default value: `false`

Note: You can use this group of options numerous times if you have more than one directory you want to package.

Example of packaging a directory:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-pdapackage -packagedir C:\Channels\PriceList  
-channelname PriceList -sourcedir  
D:\DeviceFiles\PriceListSource as \PriceList
```

Note: Command syntax and examples are sometimes shown on several lines, but you must of course type each command from the command line without any break until the end of the command.

- **CAB files.** For packaging CAB files (applications), use the following options:

-cabfile <source_CAB_file> specifies the full path to the CAB file you want to package.

unloadname <application_name> specifies the name to use to uninstall the application from the mobile device. The application name you use *must be* the same name that would appear in the Remove Programs list on your device if you installed the application without using Marimba products. Therefore, before you create your package, install the application on one mobile device without using Marimba products, and find out the name used in the Remove Programs list.

[filebyref {true|false}] means, if set to true, that the source CAB file is packaged by reference. Use this argument if you want to save disk space on the packaging machine.

Default value: false

Note: You can use this group of options numerous times if you have more than one CAB file you want to package.

Example of packaging a CAB file:

```
runchannel http://trans.acme.com:5282/ApplicationPackager  
-pdapackage -packagedir C:\Channels\MyApplication  
-channelname MyApplication  
-cabfile "D:\Device Applications\MyApp.cab"  
unloadname MyApplication
```

Tip: If you want to use the command-line interface to create a package, and then later decide to use the Application Packager PDA Packager graphical user interface (GUI) to reconfigure the package, use the Application Packager **File > Import** command in the GUI, and then browse to the channel directory. When you import the channel directory in this manner, the channel is listed in the PDA Packager.

Publishing the packages. When you successfully package the directory or CAB files by using the appropriate command-line options, use Channel Copier to actually publish the packaged application channels to your transmitter. In the Application Packager GUI, you click the Publish button to accomplish this task. (The Publishing wizard then appears and guides you through the publishing process.) If you are using the Application Packager command-line interface, however, start Channel Copier and manually copy the channels.

► **To copy a package to the transmitter**

- 1 On your packaging machine, make sure your tuner is running and then start Channel Copier as follows: Double-click the Channel Copier channel, which appears in the Channel Manager.
- 2 In the Channel Copier window that appears, click New, to create a new copy operation.
- 3 In the Select Source area, click the folder icon and then browse to the channel directory (that is, the directory that contains the contents of your new channel).

This is the directory you specified by using the `-packagedir` option when creating the PDA package.

- 4 In the Select Destination area, select the transmitter that you want to publish the channel to.
- 5 Click **Add/Close**. You are returned to the main Channel Copier window, and your new copy operation is selected.
- 6 Click **Copy** to publish your new channel.
- 7 When the copy operation has been completed, if needed, you can exit Channel Copier.

Virtual Packager command-line options

The following information describes how to use the virtual packager command-line options to package a set of directories and files:

```
runchannel <App_Packager_URL> -virtualpackage
<channel_directory><directory_list> [-defaults
<XML_template_file_path>]
```

- [-defaults <XML_template_file_path>]—specifies the XML template file to use when creating the new channel. If you omit this option, the Application Packager default XML template file is used.
- <channel_directory>—the full path of the directory to place the contents of the new channel that you are creating
- <directory_list>—the source directories

Each directory in <directory_list> has the following form:

```
<directory> [as <install_directory>]
[prompt "<text>"]
[-byreference <true|false>]
[-include <true|false>]
```

<directory>—the full path of the directory containing content to package

- as <install_directory>—the directory on the user system where the packaged files are installed when the channel is run.
- prompt "<text>"—queries the user for the directory in which to install the packaged files when the channel is run, displaying the specified text as a prompt for input. The text should be a message asking the user to specify the target directory.
- [-include]—specifies whether to include in the channel all the files that are in the same directory as the MSI file. If you omit this option, other files in that directory are not included in the channel.
- [-byreference]—specifies whether to package files by reference.

Example (Windows)

```
runchannel http://trans.acme.com:5282/ApplicationPackager
-virtualpackage C:\Channels\MyVirtualPackage C:\ContentFiles\src1
as C:\Install\src1 prompt "Where to install src1?" -byreference
true -include false
```

Upgrading channels

This command-line option upgrades channels that were packaged using a previous version of Application Packager to the current version. You can upgrade channels that were packaged with Application Packager 4.0 and later.

You can start Application Packager with the `runchannel` program and specify the channels to upgrade as follows:

```
runchannel <App_Packager_URL> -upgrade <path_of_text_file>
```

where `<path_of_text_file>` is the full path of the text file containing the directory paths of the channels to upgrade, one entry per line.

Example (Windows)

```
runchannel http://trans.acme.com:5282/Marimba/  
ApplicationPackager -upgrade C:\packagedchannels\upgrade.txt
```

where the file `upgrade.txt` contains entries like these:

```
C:\packagedchannels\app1  
C:\packagedchannels\app2  
C:\packagedchannels\app3  
C:\packagedchannels\app4  
C:\packagedchannels\app5
```

Example (UNIX)

```
runchannel http://trans.acme.com:5282/Marimba/  
ApplicationPackager -upgrade /home/user1/packagedchannels/  
upgrade.txt
```

where the file `upgrade.txt` contains entries like these:

```
/home/user1/packagedchannels/app1  
/home/user1/packagedchannels/app2  
/home/user1/packagedchannels/app3  
/home/user1/packagedchannels/app4  
/home/user1/packagedchannels/app5
```

Checking the version of Application Packager and channels

These command-line options allow you to check the version number for Application Packager and channels you created using Application Packager. These command-line options work only for the following versions:

- Application Packager version 4.7.2 and higher
- Channels packaged using Application Packager version 4.7.2 and higher

► To check the version number of Application Packager

Use the `runchannel` program and specify the URL for Application Packager as follows:

```
runchannel <App_Packager_URL> -version
```

For example:

```
runchannel http://trans.acme.com:5282/Marimba/ApplicationPackager  
-version
```

The version number, such as 4.7.2.0, appears.

► To check the version number of Application Packager that was used to package a channel

Use the `runchannel` program and specify the URL for the channel as follows:

```
runchannel <channel_URL> -version
```

For example:

```
runchannel http://trans.acme.com:5282/Applications/  
CustomApplication -version
```

The version number, such as 4.7.2.0, appears.

Applying an XML template file to channels

This command-line option applies an XML template file to an existing channel that was packaged using Application Packager. You can start Application Packager with the `runchannel` program and specify the package directory and XML template files as follows:

```
runchannel <App_Packager_URL> -applytemplate <channel_directory>  
<paths_of_XML_template_files>
```

where

`<channel_directory>` specifies the full path of the directory where you want to place the contents of the new channel that you are creating.

`<paths_of_XML_template_files>` specifies the paths of the XML template files that you want to apply to the channel. If you want to use more than one XML template file, separate the paths with semicolons (;). For more information, see the *Application Packager User Guide*.

Example (Windows)

```
runchannel http://trans.acme.com:5282/Marimba/  
ApplicationPackager -applytemplate  
“C:\Channels\MyCustomChannel”  
“C:\Channels\Templates\shrinkwrap.xml”;  
C:\Channels\Templates\policies.xml;  
C:\Channels\Templates\macros.xml”
```

Example (UNIX)

```
runchannel http://trans.acme.com:5282/Marimba/  
ApplicationPackager -applytemplate /home/user1/MyCustomChannel  
/home/user1/Channels/Templates/policies.xml;  
/home/user1/Channels/Templates/macros.xml”
```

Starting packaged applications from the command line

You can start an application packaged using Application Packager from the command line as follows:

```
runchannel <channel_URL> <options>
```

where the options can be any of those described in this section.

```
-exec [<arguments>]
```

starts the application, if any, that's configured to start from the channel, optionally passing arguments that the application has been configured to accept. (Using `-exec` without any arguments has the same effect as using `runchannel` without any options.) If the channel has not been installed, this option launches the installer first.

The `-exec` option should appear last among any options specified because everything that follows `-exec` is considered an argument to be passed to a started instance of the application.

```
-install
```

installs on the tuner the files the channel has been configured to install.

```
-refreshsourcelist
```

refreshes the source list for an MSI channel. This option is valid only for channels created using the Windows Installer Packager.

-remove

uninstalls the files the specified channel installed on the tuner.

-repair

checks the channel integrity and fixes any errors in the channel.

-rundir <directory>

specifies a new working directory for the channel.

-runexe <application_name>

starts the specified application, usually one installed by the channel.

application_name must include the full path of the application executable or batch file.

Note: You can configure the process creation associated with this command-line option by setting the `processCreationFlags` in the `parameters.txt` file of the package. For more information, see the *Application Packager User Guide*.

-setmacro <macro_name>=<macro_definition>

defines a macro for customizing channels. For example, on a Windows system it can be the following:

“-setmacro INSTALLDIR=c:\program files\app”

If the macro or macro definition includes spaces, you must enclose the entire command line within quotation marks (“”).

Note: The `-setmacro` command-line option by itself cannot cause a major update to occur when a packaged application is being updated. For example, using `-setmacro` to set change the `$dir1` macro’s value in a file-packaged channel to another value does not cause a major update to occur. For more information about major updates, see the *Application Packager User Guide*.

-silent

installs the channel in silent mode.

```
-stagems
```

stages an MSI channel. This option is valid for only channels created using the Windows Installer Packager. For more information, see “Staging MSI applications on endpoints” on page 56.

```
-verify
```

checks the channel integrity.

```
-version
```

displays the version of Application Packager that was used to create this channel. See “Checking the version of Application Packager and channels” on page 53 for more information.

The form with no options:

```
Tuner -start <channel_URL>
```

can be used to start the application, if any, that was previously installed by the specified channel (and configured to start from that channel).

Staging MSI applications on endpoints

This command-line option is useful when used with the stage state available when using Policy Management to distribute applications. You use this option as follows:

```
runchannel <channel_URL> -stagems
```

The MSI file and any support files are staged in an `msi` directory in the channel directory (for example,

`C:\<tuner_workspace_directory>\ch.X\data\msi`, where `X` is a number representing the channel).

Certificate Manager options

You can run Certificate Manager from the command line on local tuners (using `runchannel`) and remote tuners (using `tuner -start`).

Note: (**UNIX only**) Although using `runchannel` for remote certificate management is not directly supported, you can use a UNIX `telnet` command (or `ssh`) to connect to a remote UNIX workstation and invoke the commands.

Using runchannel for local certificate operations

To manage certificates on local tuners, the syntax for the command-line interface is as follows:

```
runchannel <Certificate_Mgr_URL> [<options>]
```

The Certificate Manager command-line options are described in this section. You can abbreviate each of the option names to its first letter.

-chain | -c

displays the current certificate's lineage.

-delete | -d <cert>

deletes a certificate.

-export | -e <cert> <file>

exports an installed certificate in Marimba format to a file.

-help | -h

lists and briefly describes all the Certificate Manager command-line options.

-import | -i <cert_type> <file> [<password>]

installs a certificate from a certificate file. Include the absolute or relative path with the file name. You don't need to provide the certificate's password if you are importing a root certificate.

-list | -l

displays a list of all non-root certificates installed for the tuner.

-nickname <nickname>

applies a nickname to the current certificate.

-print | -p [<cert_type>]

displays the contents of the entire certificate database or all the certificates of the specified type.

-quiet | -q

means run with little or no interaction or display.

```
-setuse | -s <use_type> <cert>
sets a root certificate's use type.
<use_type>
indicates the type of trust to assign to the specified root certificate. For
example: ssl or nossal for an SSL certificate; client or noclient for a
client certificate; or channel or nochannel for a channel-signing
certificate.
```

```
-view | -v <cert>
displays the contents of a certificate.
```

The following arguments are used with some of the Certificate Manager options:

```
<cert>
represents the certificate's distinguished name (or a part of the
distinguished name that is unique among the tuner's installed
certificates), unique ID (for example, 3R3X61-fCIQ+2-I5ZXIR-NGaQ==), or
nickname. Use quotation marks around an entry if it contains spaces or
special characters.
```

```
<cert_type>
is the type of certificate: ssl, client, code, or root.
```

For example:

```
runchannel http://trans/CertificateManager -import ssl dino-
cert.p12 bob
```

imports an SSL certificate from a file named **dino-cert.p12**; the
certificate's password is **bob**.

```
runchannel http://trans/CertificateManager -view
"cn=xena.marimba.com"
```

displays the contents of the certificate with **cn=dino.marimba.com** (the CN
portion of the DN, which is unique among the tuner's installed
certificates).

```
runchannel http://trans/CertificateManager -v 3R3X61-fCIQ+2-
I5ZXIR-NGaQ==
```

displays the contents of the certificate having the ID **3R3X61-fCIQ+2-
I5ZXIR-NGaQ==**.

Using tuner -start for remote certificate operations

Unlike runchannel, tuner -start displays the channel's user interface. Although it displays the interface on the workstation from which you issue the command, the command operates on the certificates belonging to the remote tuner.

To manage certificates on remote tuners, you run the Certificate Manager channel. The syntax for the command-line interface is as follows:

```
tuner -start <Certificate_Mgr_URL> -url <tuner_URL>:<port>
```

where

```
-url <tuner_URL>[:<port>]
```

specifies the URL and RPC port of the remote tuner. A default port number of 7717 is used if no port is specified. (Each tuner has an RPC port that was specified when the tuner was packaged.) A dialog box appears that prompts you for the remote tuner's administrative password. If you enter an invalid user name or password for the remote tuner, Certificate Manager notifies you that authentication failed, and you are able to operate on only local tuner certificates.

For example:

```
tuner -start http://trans/CertificateManager -url http://mach.marimba.com:7700
```

runs the Certificate Manager channel on trans to manage certificates on the tuner running on mach.marimba.com at RPC port 7700.

Channel Copier options

You can run Channel Copier from the command line. The syntax for the command-line interface is as follows:

```
runchannel <Channel_Copier_URL> [<options>]
```

where

```
<options>
```

specifies the Channel Copier command-line option.

The Channel Copier options are described in the rest of this section. You can specify a single source and destination for a copy operation with the `-src` and `-dst` options, or use the `-batch` option to specify a batch file for multiple copy operations.

`-batch | -b <file>`

specifies a batch file for copy operations. This option is useful for copying multiple channels from the command line. The format of the batch file is the same format that Channel Copier uses when you export a set of copy operations and save it to a file. For more information about using batch files, see Channel Copier Help.

`-clientcertpw <password>`

sets the private-key password for the client certificate. Use this option when the source transmitter, destination transmitter, or both require client certificates.

`-delete`

deletes a channel that you specify from the destination. You must include both the source and destination of the channel that you want to delete (using the `-src` and `-dst` options).

`-dst <destination> [-seg <segment_ID>] [-auth <user_name>:<password>] [-3xpw:<3xpassword> -channelurl <channel_URL>]`

where

`<destination>`

specifies the destination of the copy operation with a channel URL or a path to a CAR file.

`-seg <segment_ID>`

is the segment ID for the segment to be copied to the destination.

`-auth <user_name>:<password>`

specifies the publish authorization for the destination. You can omit it if authorization is not required for the destination.

`-3xpw:<3xpassword>`

specifies the publish authorization for a Marimba product version 3.x destination.

-channelurl <channel_URL>

specifies the channel URL from which the channel should get updates. It's important to specify the URL when the source is a directory and the destination is a CAR file because the channel may not have a URL associated with it, which can cause problems when the channel tries to update.

-help | -h

lists and briefly describes the command-line options for Channel Copier.

-paramfile <file>

specifies the file that contains the parameters to change during the copy operation.

-propsfile <file>

specifies the file that contains the properties to change during the copy operation.

-sign <cert_name> <cert_password>

signs all channels and re-signs previously signed channels. You don't need to use this option with a batch file that already includes signing information for each copy operation.

-src <source> [-seg <segment_ID>] [-auth <user_name>:<password>]

where

<source>

specifies the source with a channel URL or a path to a CAR file or a directory.

-seg <segment_ID>

is the segment ID for the segment to be copied from the source. If this option is omitted, all segments are copied.

-auth <user_name>:<password>

specifies the replication authorization for the source. You can omit it if authorization is not required for the source.

Note: Channel Copier does not support copy of Content Replicator package at the segment level. When you choose the source as a specific segment, ensure that the source is a BBCA channel or a package created using Application Packager. You can also selectively perform segment copy for a Content Replicator package by copying the entire package and then use Transmitter Administration to delete the unwanted segment in the Transmitter.

-tmp <directory>

specifies that the storage cache is created in a temporary directory. It's important to use this option when you have multiple instances of Channel Copier running at the same time — for example, when you are using scripts to execute multiple copy operations. Specifying a different storage cache directory for each copy operation ensures that the correct files are copied (and that the files are copied completely).

-unsign

unsigns any previously signed channels. You don't need to use this option with a batch file that already includes signing information for each copy operation.

-verbose | -v

changes to verbose mode and sends output to standard output unless you redirect it to a file.

Common Management Services options

You can configure some of the system settings for the CMS console and browser-based applications from the command line. The syntax of the command-line interface is as follows:

runchannel <CMS_URL> -user <user_name> -password <password> <option>

where

<CMS_URL>

is the URL of the Common Management Services channel.

<user_name>

is the user name that you use to log in to the console and to the Marimba based applications.

<password>

is the password that you use to log in to the console and to the Marimba browser-based applications. <option>

specifies a Common Management Services option.

The Common Management Services command-line options are described in the rest of this section.

-disableSSL

disables the use of SSL for the console and browser-based applications.

-enableSSL <certificate_nickname_or_ID> <certificate_password>

enables SSL for the console and browser-based applications using the specified SSL certificate and password.

Note: If you want to use encryption for the passwords, commands, and status data sent to the console, obtain and install a Secure Sockets Layer (SSL) certificate on the machine that hosts the console and browser-based applications. For more information, see the section about configuring the console to use SSL in the *Symphony Marimba Client Automation CMS and Tuner User Guide*, available on the Marimba Channel Store.

-getBindAddress

displays the bind address (usually an IP address) representing the network interface. A value of 0.0.0.0 indicates all available network interfaces.

-getLogDir

displays the path of the directory where log files for the applications are stored.

-getMaxConn

displays the maximum number of simultaneous connections allowed by the HTTP server.

-getPort

displays the port number used for accessing applications through a browser.

Default value: 8888

-getRoot

displays the path of the directory where configuration files for system settings are stored.

-getSSLCert

displays the nickname or ID of the SSL certificate that the console and browser-based applications are configured to use.

-h [<option>]

-help [<option>]

lists all the system settings command-line options. To display help for a specific command-line option, use -help <option>.

Note: Do not include the hyphen (-) when specifying the command with -help (for example: -help setPort).

-password <password>

specifies the password you use to log in to Marimba applications. (This option is required with all other options; however, if you have not set a password, you do not need to use this option.) This option, in combination with the -user option, authenticates you to Marimba applications, and enables the applications to determine whether you have primary administrator, standard administrator, or operator privileges.

-setAdminPassword <password>

specifies the emergency administrator password. The emergency administrator password allows you and other users to log in and use the applications even if the directory server (or the local user database used for authenticating users) is not available.

If you want to set the emergency administrator password to blank (no password), use quotation marks with nothing enclosed. For example:

-setAdminPassword “”. You should also use quotation marks if the password you want to specify contains spaces.

Note: Log in with the user name admin when using the emergency password.

-setBindAddress <IP_address>

specifies a bind address representing the network interface on which you want browser-based applications to accept requests. Specify a bind address only if the machine on which you are running these browser-based applications has more than one network interface. You can specify all network interfaces by entering 0.0.0.0 or just one specific network interface by entering the IP address.

```
-setDirectoryService <directory_service_name>
<directory_service_description> <directory_service_vendor>
<directory_service_hostname> <base_DN> <user_RDN> <password>
-authMethod {none|simple|strong} -useSSL {true|false}
{<primary_administrator_group>} [<administrator_group>]
[<operator_group>]
```

adds a new Active Directory Application Mode (ADAM) / Active Directory Lightweight Directory Structure (AD LDS) directory service and selects it as the active directory service. ADAM / AD LDS is the only type of directory service supported by this command. Also, this command sets the groups in the new directory service for role-based access control. For each of the group parameters, specify a single group using its CN. The primary administrator group is required. The administrator and operator groups are optional.

For example:

```
-setDirectoryService AdamTestService1 SampleAdam ADAM
"172.23.239.26:389" "dc=west,dc=us,dc=qa,dc=marimba,dc=com"
"cn=sylvio,cn=users,dc=west,dc=us,dc=qa,dc=marimba,dc=com"
marimba simple false Padmins Admins Operators
-setLogDir <directory_path>
```

specifies the directory where you want to store log files for the applications.

```
-setMaxConn <maximum_number_of_simultaneous_connections>
```

specifies the maximum number of simultaneous connections allowed by the HTTP server.

Default value: 1024

```
-setNoSSL
```

disables the use of SSL for the console and browser-based applications.

```
-setPort <directory_path>
```

specifies the port number to use for accessing applications through a browser. The port number must be an integer between 0 and 65535.

Default value: 8888

`-setRoot <directory_path>`

specifies the directory where you want to store configuration files for system settings.

`-setSSL <certificate_nickname_or_ID> <certificate_password>`

enables SSL for the console and browser-based applications using the specified SSL certificate and password.

Note: If you want to use encryption for the passwords, commands, and status data sent to the console, obtain and install a Secure Sockets Layer (SSL) certificate on the machine that hosts the console and browser-based applications. For more information, see the section about configuring the console to use SSL in the *Infrastructure Administrator's Guide*, available on the Marimba Channel Store.

`-user <user_name>`

specifies the user name to use to log in to Marimba applications. (This option is required with all other options.) This option, in combination with the `-password` option, authenticates you to Marimba applications, and enables the applications to determine whether you have primary administrator, standard administrator, or operator privileges.

The following example illustrates how you can find the path of the directory where log files for the applications are stored:

```
runchannel http://trans.acme.com:5282/cms -user john -password secret -getLogDir
```

Patch Manager options

You can run Patch Manager from the command line to perform many of its functions. The syntax of the command-line interface is as follows:

```
runchannel <PatchManager_URL> -tenant <tenant_name>  
-user <user_name> -password <password>  
<options>
```

where `<PatchManager_URL>` is the URL of the Patch Manager channel, and `<user_name>` and `<password>` are the user name and password that you use to log in to Patch Manager. The Patch Manager command-line options are described in the rest of this section. These options are grouped by the following functional areas:

- “General options” on page 65
- “Patch repository options” on page 66
- “Patch options” on page 68
- “Bulletin options” on page 69
- “Patch group options” on page 70
- “Upgrade options” on page 72

General options

`-detailedHelp`

displays detailed usage information for all commands.

`-getSystemProp <prop>`

displays the value for a system property.

`-help [<command>]`

displays usage information. If `<command>` is specified, detailed information about the command is displayed.

`-reset`

resets an environment variable instructing Patch Service to go ahead and execute the install tree, even though the tree is the identical to the one used for last attempted install.

`-setDownloadPolicy/{Windows|RedHat}/ {DownloadAll|DownloadByGroup}`

specifies the download policy. If a specific Patch Source is specified, only those patches are downloaded when the patch repository is updated. If `DownloadAll` is specified, all patches are downloaded when the patch repository is updated. If `DownloadByGroup` is specified, all patches in a patch group are downloaded when that patch group is published.

`-setSystemProp <prop> <value>`

specifies a value for the system property. If <value> is the string NULL, the property is removed and reverts to the default value.

-setUpdateSchedule <schedule>

sets the update schedule for patches. The schedule is in standard UNIX cron syntax. If the schedule is the string manual, the schedule is removed and updates must be manually started.

-showDownloadPolicy/{Windows|RedHat}

displays the current download policy for the specified Patch Source channel.

-showSystemProps

displays all available configuration properties for the system, including a description of each property.

Patch repository options

-abortUpdate [url]

cancels an ongoing update operation of the patch repository. The [url] is the URL for the Patch Source channel whose updates you want to cancel. If no Patch Source channel is specified, updates for all the Patch Source channels are canceled.

-addPatchSource <url> <tuner_url>

adds a remote patch source channel to the patch repository. The <url> is the URL for the patch source channel, which is where the collection agent on the tuner gets the patch from. The <tuner_url> is the URL of the tuner. This option prompts for a username and password from the standard input.

-delPatchSource <url>

removes the patch source channel, as specified by the URL, from the patch repository. This does not remove patches from the patch repository.

-deletePatchSourcePatches <Patch Source ID>

deletes the patches from the specified Patch Source channel. Patch Source IDs are listed in the Vendor column in the Manage patch repository dialog box. This option is useful, for example, when you want to delete custom patches from a Patch Source channel.

-dependencyFile <dest>

is for debugging purposes only. Generates a dependency file, based on the patch repository, and copies it to <dest>.

-downloadAllPatches <locales> <destination> <patchType>

downloads binaries from an Internet site and saves them to a storage location that you can use to update a Patch repository that does not have Internet access.

- locales: provide a comma-separated full locale name
- destination: full path to the target location
- patchType: specify All, Packages, or ServicePacks

Default values:

- locales: English
- destination: storage Directory of Windows Patch Source
- patchType: All

```
runchannel.exe http://10.10.51.31:5282/8000_local/pattanaik/
WindowsPatchSource -downloadAllPatches -locales English,French -
destination C:\offline -patchType All
```

-export <file>

exports all bulletins and patches in the patch repository to the selected XML file, which can then be passed into -import to insert the patches into the patch repository.

-exportPatches/{Windows|RedHat|Custom|all} <file>

Export all bulletins and patches in the selected platform or platforms in the patch repository to the selected XML file. The file can then be passed into -import to insert the patches into the patch repository. The platform value can be all to include all platforms or it can be a comma-separated list; for example, {Windows, RedHat}.

-filter {Bulletin|Patch} {Windows|RedHat|Custom}
<productVersions> <filterList>

displays a filtered list of objects in the patch repository. The <productVersions> value can be All to include all product versions or it can be a list of the form {OS|APP},<name>,<version>;.... The <filterList> parameter is a list of the form <property>,<op>,<value>,... where <property> is one of the properties returned by the -showFilters command, <op> is one of =, >, <, and <value> is the value to test for.

```
-import <file>
imports patches from the selected XML file, which must be generated with
the -export command. After you use this option to import patch data, you
need to restart Patch Manager.

-resetDatabase [force]
removes all objects from the patch repository. If force is specified, the
command does not prompt for confirmation.

-resetDatabase [force] [config]
removes all objects from the patch repository. If force is specified, the
command does not prompt for confirmation. If config is specified,
configuration information is erased; the patch information is not erased.

-showFilters {Windows|RedHat|Custom}
displays all properties that are search criteria for the -filter command.

-showMachinesByPatch <id>
lists all machines affected by the patch.

-showPatchSources
lists all currently installed patch sources, including status information
about each patch source.

-showProductVersions {Windows|RedHat|Custom}
displays all product versions that apply to patches for the specified
platforms.

-updateDatabase {[url]}|[force]| [publish]}
updates the patch repository. The <url> is the URL for the Patch Source
channel. If force is specified, the patch repository is updated, even if there
were no changes detected in the patch sources. If publish is specified, only
the patch information is published to the transmitter.
```

Patch options

```
-addCustomPatch bulletin <id> <bulletinId> [title
<title>][description <description>][type <type>][url
<url>][application <name> <version>][OS <name> <version>]
[requires <id>][conflicts <id>][obsoletes <id>][updatedOn <MM/dd/
YYYY>][keyword <keyword>][version <version>][<prop> <value>]
```

adds a custom patch to the bulletin ID which may be the string `none` if no bulletin id is applicable. The `title` property is used to provide a brief one-line summary of the patch. The `description` property is used for a more detailed explanation of the patch. If `type` is specified, it must be `Patch` (default), `Rep` (a channel produced by the Content Replicator application), or `Packaged` (a channel produced by the Application Packager product). The `url` must be specified for the patch. If the patch is a `Patch` type, this URL refers to the location of the patch binary to be packaged and is placed on the transmitter. Otherwise, this is the URL of a channel that contains the patch. If `application` or `OS` is specified, the patch applies only to the selected application or OS, which must be a product that is included in the `-showProductVersions` list. This parameter can be specified more than once to apply to multiple products or operating systems. The `requires`, `conflicts`, and `obsoletes` properties are used to list other patches that this patch has the given dependency on. The `id` given is the Marimba ID of a patch. If `updatedOn` is given then it is used as the date this patch was updated. Otherwise the current time is entered. If `keyword` is given then it is an arbitrary string that is added to the keyword section for this patch. This property can be specified more than once to associate multiple keywords with the patch. The `version` property is used to enter a version for this patch. In addition an arbitrary list of property value pairs may be specified. The ID of the new patch is included.

`-packagePatch <id> [<url>]`

packages the selected patch. If the URL is specified, the patch is downloaded from that URL and packaged. Otherwise, the patch is downloaded from its original location, if it's known.

`-patchSummary`

prints a list of all patches, including a brief summary of each patch.

`-showPatch <id>`

displays the patch, including the database ID.

Bulletin options

`-addCustomBulletin <id> [title <title>] [description <description>] [<prop> <value>]`

adds a custom bulletin to the patch repository. The ID of the new bulletin is included.

-
- bulletinSummary
 - prints a list of all bulletins, including a brief summary of each bulletin.
 - showBulletin <*id*>
 - displays the bulletin, including the database ID.

Patch group options

- abortPublish <*name*>
 - stops the publishing operation of a patch group.
- addPatchGroupScript <*patch group*> <*name*> <*file*>
 - adds the script to a patch group. The script is copied from the absolute file path.
- addPatchToPatchGroup <*name*> <*id*> [<*prop*> <*value*>]
 - adds the patch to a patch group. A list of property/value pairs may be added. If the patch already exists in the patch group, its properties are replaced (overwritten) by the properties specified in this command.
- addQueryToPatchGroup <*groupName*> {Windows | Custom} <*productVersions*> <*filterList*>
 - adds a SQL query, derived from specified arguments, to a dynamic patch group. If *groupName* already refers to a dynamic patch group, the query is replaced. If *groupName* already refers to a static patch group, the command reports an error. The filter arguments are the same as the -filter command.
- copyPatchGroup <*name*> <*new_name*> [<*description*>]
 - copies the patch group referred to by <*name*>. The copy of the patch group includes the <*new_name*> and <*description*>.

```
-createPatchGroup <name> <type> <description> [<installRecursive>  
<uninstallRecursive>]
```

creates a patch group called `<name>`. If the patch group already exists, the specified options are applied to the existing patch group, excluding the type, which can't be changed. The type parameter is either static or dynamic. A dynamic patch group is associated with a database query, whereas a static patch group is explicitly constructed by invoking the addPatchToPatchGroup option. The installRecursive option is either true or false, where pre-required patches are automatically installed, even if they are not part of the patch group. The uninstallRecursive option specifies that any patch that depends on a patch that is being uninstalled is also uninstalled. The default for uninstallRecursive is false.

```
-delPatchFromPatchGroup <name> <id>
```

deletes the patch from a patch group.

You cannot delete patches from a patch group that is being published. Either stop the publish or wait until the publish is done.

```
-delPatchGroup <name>
```

deletes the patch group and remove it from the transmitter.

```
-forcePublish <-bulletin><-patch>, where <-bulletin> is the bulletin  
number and <-patch> is the patch ID
```

republishes a published bulletin or patch that Microsoft has updated. You can specify only one argument.

```
-forcePublishPatchGroup <-group_name>, where <-group_name> is the  
name of the patch group
```

republishes a patch group that contains a published patch that Microsoft has updated.

```
-getPatchGroupScript <patch group> <name> <file>
```

copies the script from the patch group to a specified file path.

```
-publishPatchGroup <name> [force]
```

publishes the patch group to the transmitter. If force is specified, all patches in the patch group are republished to the transmitter, regardless of whether they were previously published.

-
- revertPatchGroup <name>
 - reverts the selected patch group to the action it was in when it was published. If the patch group was never published, an error is returned.
 - showPatchGroup <name>
 - displays detailed information about the patch group.
 - showPatchGroups
 - lists all currently defined patch groups.
 - showPatchProps
 - lists all properties that can be defined for a patch (which are added to a patch group), including a description of each property and the default value (if the property is not specified in the addPatchToPatchGroup command).
 - simulate <host> <patch group urls>
 - displays a simulated chronology of what actions would occur if the selected patch groups (list of URLs) were deployed to an endpoint.

Upgrade options

You can use these options when upgrading from version 2.x and earlier to the current version.

- clearPatchEdits <file>
 - reverts to the vendor defaults for patch edits in the specified file.
- clear20PatchEdits <file>
 - reverts to the vendor defaults for version 2.x and earlier patch edits in the specified file.
- exportPatchEdits <file>
 - exports patch edits to the specified XML file. You can then use -importPatchEdits to insert the patch edits into the patch dictionary.
- export20PatchEdits <file>
 - exports version 2.x and earlier patch edits to the specified XML file. You can then use -importPre65PatchEdits to insert the patch edits into the current patch dictionary.
- importPatchEdits <file>
 - imports patch edits from the specified XML file.

- import20PatchEdits <file>
 - imports the version 2.x and earlier patch edits from the specified file and save them to the current version.
- upgradePatchEdits
 - upgrades the version 2.x and earlier patch dictionary to the current patch dictionary.
- upgradePatchRepository
 - upgrades the version 2.x and earlier patch repository to the current patch dictionary by upgrading patch edits, patch groups, bookmarks, and custom patches.

Patch Source options

You use these options for obtaining patches from the sources for the operating systems and programs.

Patch Service options

You use these options primarily for server management. Because there is no update schedule for servers, these options let you work with all patches at once.

- forceInstall
 - sets all patch actions in each patch group to install for the duration of the install.
- forceRemove
 - sets all patch actions in each patch group to uninstall for the duration of the uninstall.
- forceStage
 - sets all patch actions in each patch group to stage for the duration of the staging.
- install
 - performs the specified actions (install, uninstall, stage, do nothing) for the patch groups that are on a tuner. There is no channel update.

-preview

installs patch groups (including setting dependencies and patch order) but don't perform the action. This option is used for debugging.

-updateInstall

performs a channel update of the Patch Service channel and then perform the specified actions (install, uninstall, stage, do nothing) for the patch groups.

Policy Manager options

You can run Policy Manager from the command line to configure Policy Manager and use it to create and assign policies. The syntax of the command-line interface is as follows:

```
runchannel <PolicyManager_URL>
-user <user_name> -password <password>
<options>
```

where *<PolicyManager_URL>* is the URL of the Policy Manager channel, and *<user_name>* and *<password>* are the user name and password that you use to log in to Policy Manager. The Policy Manager command-line options are described in the rest of this section.

```
-changeorder {<target_name> <target_type> | -dn
<distinguished_name>} [<package_url>]=[package_priority_number]}
```

modifies the install priority of packages in a Policy by changing numeric values associated with the packages, enabling you to override the original installation sequence the policy specified. If another package in the policy currently owns a package priority number you assign, the system increments the other package's number by one. After assigning each specified package to a new priority, the procedure then renames all packages on the priority list to make the numeric values sequential.

where

<target_name> specifies the name of the target. Depending on the target type (value of *<target_type>*), this argument refers to a machine name in the directory service, or to a user name or a group name obtained either from the directory service or from the transmitter.

<target_type> specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

- user
- usergroup
- machine
- machinegroup
- all

If <target_type> is all, the system ignores <target_name> and the target object is given the name all_all.

Note: *Collection* is not a valid argument in the command-line interface. Use the machinegroup target type to designate a collection.

<distinguished_name> specifies the distinguished name (DN) of the target in the directory service.

<package_url> specifies the URL of the package whose install priority you want to change.

The following example demonstrates how to use the -changeorder flag:

```
C:\Program Files\Marimba\Tuner>runchannel http://
10.10.137.137:5282/john/SubscriptionManager
-user john
-password opensesame
-changeorder computers container http://alfonzo:80/luser/
Publisher=5
-clientcertpw <password>
```

specifies the password for the client certificate used when publishing the Policy Service plug-in to the transmitter. You usually use it with the -publish option.

where

<password> specifies the client-certificate password.

The following example shows -clientcertpw used with the -publish option:

```
runchannel http://trans.company.com:5282/SubscriptionManager
-user john -password opensesame
-clientcertpw companycert
-publish http://trans.company.com:5282/SubscriptionService
-configSet <key> <value> [-preview}
```

sets the specified Policy Management configuration property. For more information, see the section about the Subscription Config object in the *Policy Management User Guide*, available from the Marimba Channel Store.

where

<key> <value> is the key name and value for the attribute in the subscription configuration object that you want to change.

{-preview} allows you to view the new and old values for the attribute.

Note: Although the command-line interface can tell you it succeeded when using the -preview option, it has not made the attribute change. Run the command without -preview to actually make the change.

Example:

```
runchannel http://mycompany:5282/SubscriptionManager  
-user john -password opensesame  
-configSet marimba.subscription.acl true
```

This example allows you to turn on the ACL feature by setting the value of the marimba.subscription.acl attribute to true.

-D {<dn> | <uid>}

(deprecated; see -user {<dn> | <cn> | <uid> | <sAMAccountName> | <upn>}) specifies the name of the user logging in to use Policy Manager. You can use either the distinguished name (DN) or user ID. Together with the password, the user name is used to authenticate users before executing Policy Manager commands.

where

<dn> specifies the name of a directory service user in the distinguished name (DN) format. For example, you can specify an Active Directory user as cn=john,cn=users,dc=mycompany,dc=com or a Oracle Directory Server user as uid=john,ou=People,dc=mycompany,dc=com.

<uid> specifies the user ID or login name of a directory service user. For example, you can specify an Active Directory user in the following formats:

- The user principal name (UPN) such as john@mycompany.com
- The logon name (sAMAccountName) such as john

or a Oracle Directory Server user with the user ID such as john.

```
-delete {<target_name> <target_type>}|
{<policy>} |
-all |
-cascade |
-dn <dn>}
```

deletes one or more policies. The command has five variations:

- -delete <target_name> <target_type> deletes the policy specified by the target name and target type from the current child container in the directory service.

where

<target_name> specifies the name of the target. Depending on the target type (value of <target_type>), this argument refers to a machine name in the directory service or to a user name or a group name obtained either from the directory service or from the transmitter.

<target_type> specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

```
user
usergroup
machine
machinegroup
all
```

Note: *Collection* is not a valid argument in the command-line interface. Use the `machinegroup` target type to designate a collection.

- -delete {<policy>} deletes one or more policy entries, specified using <policy> arguments from the current child container in the directory service.

The name of the policy object is obtained by concatenating the two strings <target_name> and <target_type>, separating them by an underline character (_). For example, the following command identifies the policy for the user john:

```
-delete john_user
```

To provide backward compatibility, the name of the policy can be a file name, such as:

-delete john_user.sub smith_machine.sub eng_usergroup.sub

- -delete -all deletes all policies from the current child container in the directory service.

- -delete -cascade deletes all policies from the directory service.

- -delete -dn <dn> deletes the policy specified by the target's distinguished name (DN). This command is useful when two or more groups have the same common name under different organizational units in the directory service, such as

cn=salesgroup,ou=newyork,dc=mycompany,dc=com and

cn=salesgroup,ou=sanfrancisco,dc=mycompany,dc=com.

-export <directory>

exports all the policies as policy files (.sub files) to the specified directory.

where

<directory> specifies the full path and name of a directory (folder) on the local file system.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionManager
```

```
-user john -password opensesame
```

```
-export "c:\subscription_files"
```

-import (<directory> | {<file>})

imports one or more policy files (.sub files) that have been exported from Policy Manager.

where

<directory> specifies the full path and name of a directory (folder) on the local file system that contains one or more policy files to be imported. If some objects in the imported files already exist as objects in the directory service, the import fails.

<file> specifies the full path and name of a file that contains policy information, with the format <target_name>_<target_type>.sub. You can specify multiple policy files, separating them with a space. If some objects in the imported files already exist as objects in the directory service, the import fails.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-import "c:\subscription_files\v47_policies.sub"
```

```
-ldapservers <mapping_file>
```

imports transmitter-to-directory service mappings from the specified file. These mappings are used to assign a list of directory services to each repeater.

Note: There is no GUI interface in the current release for this feature. The mapping file can be specified from only the command line.

where

<*mapping_file*> is the full path and name (such as c:\ldap\map_file.txt) of the file that lists the directory services. This file has the following format:

```
<transmitter_name>,server=<server_value>
<transmitter_name>,basedn=<basedn_value>
<transmitter_name>,binddn=<binddn_value>
<transmitter_name>,password=plain:<password_value>
<transmitter_name>,usessl=<ssl_value>
```

where

<*transmitter_name*> is the machine name or IP address of the machine on which the transmitter is running.

<*server_value*> is a comma-separated list of one or more <*host*>:<*port*> values (such as machine1:389,machine2:389). If you specify more than one <*host*>:<*port*> value, the list of servers is used for failover.

Note: Each server or repeater in the list must be configured to use the same <*basedn_value*>, <*binddn_value*>, <*password_value*>, and <*ssl_value*> settings.

<*basedn_value*> is the distinguished name (DN) of a container in the directory service (such as dc=company,dc=com)

<*binddn_value*> is the distinguished name (DN) of the user. This value is used by the Policy Service plug-in to connect to the directory service.

<*password_value*> is the password in plain-text (unencoded).

Note: (Update) The current version of Policy Manager recognizes the Base64-encoded password saved by Policy Management version 4.7.x in the LDAP server mapping file.

`<ssl_value>` determines if the plug-in connects to the directory server in SSL mode. If true, the plug-in tries to connect the directory server in secure (SSL) mode.

The following example shows the contents of a mapping file:

```
mytransmitter,server=myldap:389  
mytransmitter,basedn=dc\=mycompany,dc\=com  
mytransmitter,binddn=uid\=jouhn,ou=\people,dc\=mycompany,dc\=com  
mytransmitter,password=plain:opensesame  
mytransmitter,usessl=false
```

Directory services are typically replicated across an organization to improve response and minimize network traffic. You can take advantage of replicated directory services by configuring Policy Management so that each repeater contacts a nearby directory service, eliminating the need to contact the one assigned to the master transmitter. Moreover, you can assign a list of directory services to each repeater. If one directory service in the list fails, the repeater attempts to contact the next one, eliminating single point of failure problems. The mechanism for mapping directory services to repeaters is the LDAP mapping file. In the mapping file, you associate each repeater name with a list of directory services identified by host name, port number, base DN, bind DN, and password.

Note: For the changes you make with this command to take effect, you must publish them to the Policy Service plug-in. You can publish the plug-in changes in the same session in which you use the `-ldapservers` command. For more information, see the `-publish` command.

Example:

```
runchannel http://trans.mycompany.com:5282/SubscriptionManager  
-user john -password opensesame  
-ldapservers "c:\ldap\map_file.txt"
```

```
-list [<target_name> <target_type> |  
{<policy>} |  
-cascade |  
-channel <channel_URL> |  
-dn <dn> ]
```

lists information about one or more policies. The command has five variations:

- -list (without any arguments) lists attributes of all policies in the current child container of the directory service.
- -list <target_name> <target_type> lists the policies specified by the target names and target types in the current child container of the directory service.

where

<target_name> specifies the name of the target. Depending on the target type (value of <target_type>), this argument refers to a machine name in the directory service or to a user name or a group name obtained either from the directory service or from the transmitter.

<target_type> specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

user
usergroup
machine
machinegroup
all

If <target_type> is all, <target_name> is ignored and the target object is given the name all_all.

Note: *Collection* is not a valid argument in the command line interface. Use the machinegroup target type to designate a collection.

- -list {<policy>} lists attributes of the specified policy objects. Only objects from the current child container (specified with -namespace) are listed.

where

<policy> specifies a policy object. The value of *<policy>* must be of the form *<target_name>_<target_type>.sub* (such as *john_user.sub*) for backward compatibility with earlier Policy Management releases.

- -list -cascade lists all policies present in the directory service.
- -list -channel *<channel_URL>* lists the subscription targets for which a channel/package is currently assigned. For example:
 - runchannel http://10.10.51.56:5282/rtgadmin/
SubscriptionManager -user z -password z -list -channel http://10.10.51.20:80/CCMBDDM/712a_ga/BDDMService
 - If the channel url contains spaces, you must specify the url within double quotes. For example:

runchannel http://10.10.51.56:5282/rtgadmin/
SubscriptionManager -user z -password z -list -channel "http://10.10.51.20:80/CCMBDDM/712a_ga/BDDM Service"
 - -list -dn *<dn>* lists the policy specified by the target's distinguished name (DN). This command is useful when two or more groups have the same common name under different organizational units in the directory service, such as
cn=salesgroup,ou=newyork,dc=mycompany,dc=com and
cn=salesgroup,ou=sanfrancisco,dc=mycompany,dc=com.

-machines *<machines_file>*

imports the specified machines flat file into Oracle Directory Server.
Machines information cannot be imported into Active Directory.

where

<machines_file> is the full path and name (such as c:\ldap\mac_file.txt) of the file that lists the machines and machine groups. This file has the following format:

*<machine_name1>:<group_name1>,<group_name2>,...
<machine_name2>:*

where

<machine_name1> is a machine to be created in the directory service.

<group_name1> and *<group_name2>* are names of groups to which *<machine_name1>* belongs.

<machine_name2> is a machine that is not part of any group; no group names follow the colon (:).

If a machine name specified in the file already exists in the directory service, Policy Manager does not overwrite the existing machine object. Policy Manager adds the machine object to the specified groups (if any) if the machine is not already a member of that group.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-machines "c:\ldap\mac_file.txt"
```

-namespace <child_container>

specifies the child container (previously called namespace) used for storing policies in the directory service. You use this command-line option with other options.

where

<child_container> is the common name (CN) attribute of a container object used to store policy entries. This container is used for listing, creating, and deleting policies. It is also used to set tuner properties for Policy Management entries residing directly under the <child_container> container object.

If the -namespace option is not specified, Policy Manager uses the top-level container specified by the Subscription configuration object in the directory service.

Note: Policy Manager supports only one level of child containers under ou=Subscription <Suffix>.

In the following example, Policy Manager lists the policies in the child container child1:

```
runchannel http://trans.company.com:5282/SubscriptionManager
-user john -password opensesame
-namespace child1
-list
```

-password <password>

specifies the password of the user logging in to use Policy Manager. Together with the user name, the password is used to authenticate users before executing Policy Manager commands. It is also used when connecting to the directory service.

```
-patchsubscribe
[-modify]
{<target_name> <target_type> | -dn <dn>}
{<patchgroup_URL>=<assignment_state>,[<exempt_from_blackout>]}
[-schedpatch <date_time_range_frequency>]
[-noautoreboot|-autoreboot
[noalert|countdown=<countdown_minutes>,[postpone=<postpone_minutes>]]]
```

subscribes a target, identified by <target_name> and <target_type> or by <dn>, to a patch group identified by <patchgroup_URL>.

where

`-modify` specifies that you want to edit an existing policy, but you do not want to overwrite it; changes and additions that you make are appended to the policy. If you omit the `-modify` option, your changes overwrite any previously assigned packages, patch groups, schedules, and settings in an existing policy.

`<target_name>` specifies the name of the target. Depending on the target type (value of `<target_type>`), this argument refers to a machine name in the directory service, or to a user name or a group name obtained either from the directory service or from the transmitter. Policy Manager does not create a policy object if the user or machine or group specified by `<target_name>` does not exist.

`<target_type>` specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

- `user`
- `usergroup`
- `machine`
- `machinegroup`
- `all`

If `<target_type>` is `all`, `<target_name>` is ignored and the target object is given the name `all_all`.

Note: `Collection` is not a valid argument in the command-line interface. Use the `machinegroup` target type to designate a collection.

`-dn <dn>` specifies the distinguished name (DN) of the target. This option is useful when two or more groups have the same common name under different organizational units (OUs) in the directory service, such as `cn=salesgroup,ou=newyork,dc=mycompany,dc=com` and `cn=salesgroup,ou=sanfrancisco,dc=mycompany,dc=com`.

`<patchgroup_URL>` specifies the URL of the patch group that you want to distribute to targets. You can assign multiple patch groups to a target by specifying multiple patch group URLs.

`<assignment_state>` determines whether or not a patch group should be assigned to target machines. There are two patch group assignment states that you can assign to patch groups in Policy Manager: assign or exclude. For more information about the assignment states, see the *Policy Management User Guide*, available on the Marimba Channel Store. If you assign multiple patch groups to a target, you must specify an assignment state for each one.

`<exempt_from_blackout>` specifies whether or not the patch group identified by `<patchgroup_URL>` should be exempt from the blackout period. True indicates that the patch group is exempt from the blackout period, while false indicates the blackout period applies to the patch group. If you assign multiple patch groups to a target, you can specify a blackout exemption option for each one.

`-schedpatch <date_time_range_frequency>` specifies the schedule for updating Patch Service on a target. During an update, Patch Service scans the target for the list of required and already installed patches. It also installs a set of patches, taking into account the dependencies of each patch. You can specify that updates take place on a recurring schedule. This schedule applies to the entire target; you cannot specify a different schedule for each patch group.

For more information about the format for the schedule, see “Format for the update or repair schedule” on page 102.

`-autoreboot` or `-noautoreboot` specifies whether or not you want the target machine to automatically reboot after installing any patches that require a reboot. This option and the ones that follow applies to the entire target; you cannot specify different reboot options for each patch group.

`noalert` specifies that you do not want the target machine to display alert messages warning users about the reboot.

`countdown=<countdown_minutes>` specifies the number of minutes to wait before rebooting the machine. The reboot countdown gives users some time to save their work before the machine reboots.

`postpone=<postpone_minutes>` specifies the total number of minutes you want to allow users to postpone a reboot. After this number of minutes, users can no longer postpone a reboot, and the reboot countdown starts.

The following example subscribes the user `john` to the patch group `MyPatchGroup` with the assignment state `assign`. The patch group is exempted from the blackout period. It sets the update schedule for Patch Service to every two weeks on Mondays and Wednesdays at 4 AM. It also specifies that the target machines automatically reboot after installing any patches that require a reboot. Users can postpone the reboot up to 5 minutes, and the machine waits (count down) 30 seconds before rebooting.

```
runchannel http://trans.mycompany.com:5282/SubscriptionManager  
-user john -password opensesame  
-patchsubscribe -modify john user  
http://trans.mycompany.com:5282/MyPatchGroup=assign,true  
-schedpatch "every 2 weeks on mon+wed update at 4:00AM"  
-autoreboot countdown=30,postpone=5  
-publish <PolicyService_URL>
```

publishes the Policy Service plug-in to a transmitter.

where

<*PolicyService_URL*> specifies the transmitter host name, port and Policy Service plug-in location where the plug-in is published.

<*PolicyService_URL*> must end with the string `SubscriptionService`, such as `http://trans.company.com:5282/SubscriptionService`. If you do not provide the `SubscriptionService` string, it is automatically appended to the URL.

If access control is enabled on the transmitter, you can provide the required password (and user name, if necessary) using the `-publishpw` command. The `-publishpw` command is ignored if access control is not enabled on the transmitter.

The arguments to `-publish` and `-publishpw` are not stored by Policy Manager, so they must be specified each time a publish operation is performed.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-publish http://trans.company.com:5282/SubscriptionService
```

```
-publishpw <user_name> <password>
```

specifies the user name and password (if necessary) required for publishing the Policy Service plug-in to a transmitter. It is usually used with the `-publish` option. The `-publishpw` command is ignored if access control is not enabled on the transmitter.

where

`<user_name>` specifies the name of the user allowed to publish to the transmitter. This argument is required only if the transmitter access control setting is based on a user name and a password. If the transmitter access control requires only the password, the `-publishpw` command requires only the `<password>` argument.

`<password>` specifies the password of the user allowed to publish to the transmitter, which is specified in plain text (unencoded).

Note: The user name and password that you specify with this option are the ones required when publishing to the transmitter. You still need to specify the user name and password required for authentication by the Policy Manager (see the `-user` and `-password` command-line options).

The following example shows how the `-publishpw` option can be used with the `-publish` option:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-publish http://trans.company.com:5282/SubscriptionService  
-publishpw john txpublishpw  
-reporter <parameter_list>
```

invokes the Policy Service in reporter mode and lists the policies assigned to an endpoint and user. For example, if you assign Channels A through C to an endpoint, then running this command lists Channel A, Channel B, and Channel C. If you run this command without using `-machinename` to identify an endpoint, the reporter lists the policies assigned to your local host.

where

`<parameter_list>` can include any combination of the following options:
`-v` indicates verbose mode.

-machinename <machinename> provides the machine name for which you want to list the policy.

-userdn <userdn> specifies the fully qualified distinguished name (DN) for a user target. Use this option only for Active Directory (Auto Discovery Mode).

-machinedn <machinedn> specifies the fully qualified DN for a machine target. DNs are useful for simulating plug-in and targets that are not in the same domain in an Active Directory forest environment. Use this option only for Active Directory (Auto Discovery Mode).

-d <name> specifies the name of the directory used for storing the data obtained from the plug-in.

-refreshconfig <refreshconfig> ensures that Policy Reporter does not cache the config.xml file and does not show inaccurate results if the plug-in cannot contact the directory service.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionService  
-reporter -machinename TestMachine1  
-setpluginparam  
-pbnddn <bind_dn> -bindpasswd <password>  
-pbasedn <base_dn>  
-poolsize <pool_size>  
[-usessl]  
{-phost <host>:<port>}  
[-expirytime <expiry_time_for_last_successful_host_in_minutes>]
```

sets parameters used by the Policy Service plug-in to connect to the directory service. You must specify the -publish <PolicyService_URL> command, along with the -setpluginparam command, to publish the Policy Service plug-in with the new parameters.

where

<bind_dn> and <password> specifies the distinguished name (DN) and password of the user account used when the Policy Service plug-in establishes a connection to the directory service. This user account should have read permissions in the scope of the directory defined by the <base_dn>, which typically maps to the entire directory tree, such as cn=john,ou=users,dc=mycompany,dc=com. For Active Directory, the user ID can be entered using user principal name (UPN) format, such as john@mycompany.com.

<base_dn> specifies the base distinguished name (DN) for the Policy Service plug-in's directory connection. The *<base_dn>* determines the scope of the directory view, as seen by the plug-in. In most cases, the *<base_dn>* is equivalent to the Oracle Directory Server suffix or the Active Directory domain name, such as `o=mycompany.com`.

<pool_size> sets the maximum number of connections established in the pool used by the Policy Service plug-in to establish and maintain connections to the directory service. Typically, it can be left set to the default value of 25.

`[-usessl]` specifies that an SSL (secure sockets layer) connection should be used when publishing the plug-in.

<host>:<port> specifies the host name and port number of a directory service. You can enter multiple servers in the form *<host>:<port>* to create a server list to provide failover protection, such as `server1:389,server2:389`.

The plug-in tries to connect to each directory service in the list, in succession. If it cannot connect to a particular directory service, it then attempts to connect to the next server in the list until it succeeds or exhausts the list. Server failover functions for both initial connections and also during normal plug-in operation. As long as there is another available directory service in the list, failover should be seamless.

Note: This functionality does not provide load balancing capabilities.

<expiry_time_for_last_successful_host_in_minutes> specifies the expiration time for a directory service connection. If you specify a list of host names for directory services failover, Policy Manager goes down the list until it successfully connects to a host. It uses that successful host connection until the expiration time that you specify. Then, Policy Manager attempts to make a new connection to the first host name in the list. Make sure this expiration time follows the host name and port number.

Example:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-setpluginparam -pbinddn“cn=Directory Manager” -bindpasswd  
password -poolsize 25 -phost ldap_server1:389 -expirytime 10
```

```
-subscribe  
[-modify]  
{<target_name> <target_type> | -dn <dn>}  
{<package_URL>=<package_state1>,[<package_state2>],[<package_priority_number>], [<exempt_from_blackout>]}  
[-schedblackout <time_range>]  
[-schedprimary {<package_URL>=<date_time_range>}]  
[-schedsecondary {<package_URL>=<date_time_range>}]  
[-schedupdate {<package_URL>=<date_time_range_frequency>}]  
[-schedverifyrepair {<package_URL>=<date_time_range_frequency>}]
```

subscribes a target, identified by `<target_name>` and `<target_type>` or by `<dn>`, to a package identified by `<package_URL>`.

where

`-modify` specifies that you want to edit an existing policy, but you do not want to overwrite it; changes and additions that you make are appended to the policy. If you omit the `-modify` option, your changes overwrite any previously assigned packages, schedules, and settings in an existing policy.

`<target_name>` specifies the name of the target. Depending on the target type (value of `<target_type>`), this argument refers to a machine name in the directory service, or to a user name or a group name obtained either from the directory service or from the transmitter. Policy Manager does not create a policy object if the user or machine or group specified by `<target_name>` does not exist.

`<target_type>` specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

- user
- usergroup
- machine
- machinegroup
- all

If `<target_type>` is `all`, `<target_name>` is ignored and the target object is given the name `all_all`.

Note: *Collection* is not a valid argument in the command-line interface. Use the `machinegroup` target type to designate a collection.

-dn <dn> specifies the distinguished name (DN) of the target. This option is useful when two or more groups have the same common name under different organizational units (OUs) in the directory service, such as cn=salesgroup,ou=newyork,dc=mycompany,dc=com and cn=salesgroup,ou=sanfrancisco,dc=mycompany,dc=com.

<package_URL> specifies the URL of the package that you want to distribute to targets.

<package_state1>,[<package_state2>] specifies the primary and secondary states for the package identified by <package_URL>. The following states are available: (For more information about the package states, see the *Policy Management User Guide*, available from the Marimba Channel Store.)

- advertise
- exclude
- install
- install-persist
- install-start
- install-start-persist
- primary
- stage
- uninstall

<package_priority_number> is an integer that specifies the installation priority for the package identified by <package_URL>.

<exempt_from_blackout> specifies whether or not the package identified by <package_URL> should be exempt from the blackout period. True indicates that the package is exempt from the blackout period, while false indicates the blackout period applies to the package.

-schedblackout <time_range> specifies the blackout period for the target. For more information about the format for the schedule, see “Format for the blackout schedule” on page 99.

-schedprimary {<package_URL>=<date_time_range>} specifies the schedule for enforcing the primary installation state of the package. For more information about the format for the schedule, see “Format for the primary or secondary schedule” on page 102.

-schedsecondary {<package_URL>=<date_time_range>} specifies the schedule for enforcing the secondary installation state of the package. For more information about the format for the schedule, see “Format for the primary or secondary schedule” on page 102.

-schedupdate {<package_URL>=<date_time_range_frequency>} specifies the update schedule for the package. For more information about the format for the schedule, see “Format for the update or repair schedule” on page 102.

-schedverifyrepair {<package_URL>=<date_time_range_frequency>} specifies the repair schedule for the package. For more information about the format for the schedule, see “Format for the update or repair schedule” on page 102.

The following example subscribes the user `john` to the package `MyPackage` with the primary state `install` and the priority 1. It sets a primary schedule that is active between January 1, 2002 at 4 AM and June 30, 2002 at 6 PM and an update schedule that updates every two weeks on Mondays and Wednesdays at 4 AM, active beginning at 5 AM on January 1, 2002:

```
runchannel http://trans.mycompany.com:5282/SubscriptionManager  
-user john -password opensesame  
-subscribe -modify john user  
http://trans.mycompany.com:5282/MyPackage=install,1  
-schedprimary http://trans.mycompany.com:5282/  
MyPackage="active 01/01/2002@4:00AM - 06/30/2002@6:00PM"  
-schedupdate http://trans.mycompany.com:5282/MyPackage="every 2  
weeks on mon+wed update at 4:00AM active 01/01/2002@5:00AM"  
-tuner {<target_name> <target_type> | -dn <dn>}  
{<property_name>[,<property_type>]=<property_value>}
```

sets one or more properties for a target tuner specified by `<target_name>` and `<target_type>` or by `<dn>`. This command-line option can also be used to set one or more properties for one or more packages on the target tuner. For more information about tuner and package properties, see the *Policy Management User Guide*, available from the Marimba Channel Store.

where

`<target_name>` specifies the name of the target. Depending on the target type (value of `<target_type>`), this argument refers to a machine name in the directory service, or to a user name or a group name obtained either from the directory service or from the transmitter.

<target_type> specifies the type of the target. This argument refers to the type of the object for which the policy is being created. The following values are recognized target types:

- user
- usergroup
- machine
- machinegroup
- all

If *<target_type>* is all, *<target_name>* is ignored and the target object is given the name all_all.

Note: *Collection* is not a valid argument in the command-line interface. Use the machinegroup target type to designate a collection.

-dn *<dn>* specifies the distinguished name (DN) of the target. This option is useful when two or more groups have the same common name under different organizational units (OUs) in the directory service, such as cn=salesgroup,ou=newyork,dc=mycompany,dc=com and cn=salesgroup,ou=sanfrancisco,dc=mycompany,dc=com.

<property_name> is the name of the property you want to set.

<property_type> specifies the type of property you want to set: * (all packages), subscribers (all subscribed packages), *<package_url>* (a specific package), service (Policy Service channel). Omit *<property_type>* for tuner properties.

<property_value> is the value of the property you want to set.

For more information about the format for setting properties, see the *Policy Management User Guide*, available from the Marimba Channel Store.

The following example sets the following properties for the user john:

- The tuner property marimba.security.trusted.transmitters is set to the value trans.company.com.
- The package property reboot.showdialog is set to true for the package MyPackage.

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-tuner john user  
marimba.security.trusted.transmitters=trans.company.com  
reboot.showdialog,http://trans.company.com:5282/MyPackage=true  
-txadminaccess <user_name> <password>
```

specifies the user name and password for transmitters with restricted access. The user name and password is used to authenticate against the transmitter. For example, if Policy Manager sources users and user groups from the transmitter, the user name and password are used when obtaining the user and user group lists from the transmitter.

where

<*user_name*> specifies the name of the user who has access to the transmitter. If the transmitter is configured to use a password only for administration, enter “*” for <*user_name*>.

<*password*> specifies the password of the user with transmitter access, specified in plain text (unencoded). If you want to set the password to blank (no password), use quotation marks with nothing enclosed, such as -txadminaccess john “”. You should also use quotation marks if the password you want to specify contains spaces.

Note: The user name and password that you specify with this option are the ones required when subscribing to packages on the transmitter. You still need to specify the user name and password required for authentication by the Policy Manager. For more information, see the -user and -password command-line options.

The following example shows how you can use -txadminaccess to specify the user name and password for a restricted transmitter, so that you can subscribe the user john to the package MyPackage:

```
runchannel http://trans.company.com:5282/SubscriptionManager  
-user john -password opensesame  
-subscribe john user  
http://trans.company.com:5282/MyPackage=install  
-txadminaccess john subscribepw
```

`-user {<dn> | <cn> | <uid> | <sAMAccountName> | <upn>}`

specifies the name of the user logging in to use Policy Manager. Together with the password, the user name is used to authenticate users before executing Policy Manager commands. It is also used when connecting to the directory service.

The user name can be in the following formats:

- `<dn>` specifies the name of a user in the distinguished name (DN) format. For example, you can specify an Active Directory user as `cn=john,cn=users,dc=mycompany,dc=com` or a Oracle Directory Server user as `uid=john,ou=People,dc=mycompany,dc=com`.
- `<cn>, <uid>, or <sAMAccountName>` specifies the common name (CN), user ID, or logon name (sAMAccountName) of a user. For example, you can specify an Active Directory user with the logon name `john` or a Oracle Directory Server user with the user ID `john`.
- `<upn>` specifies the user principal name (UPN). For example, you can specify an Active Directory user with the UPN `john@mycompany.com`

`-w <bind_password>`

(deprecated; see `-password`) specifies the directory service bind password, specified in clear text without encoding. Together with the user name, the password is used to authenticate users before executing Policy Manager commands.

Schedule formats

This section describes the schedule formats that you use to specify a schedule when you use the `-subscribe` command.

Format for the blackout schedule

You use the following command-line option to specify a blackout schedule:

`-schedblackout <time_range>`

where

`<time_range>` is a range of time in the form

`HH:MM{AM|PM} - HH:MM{AM|PM}`

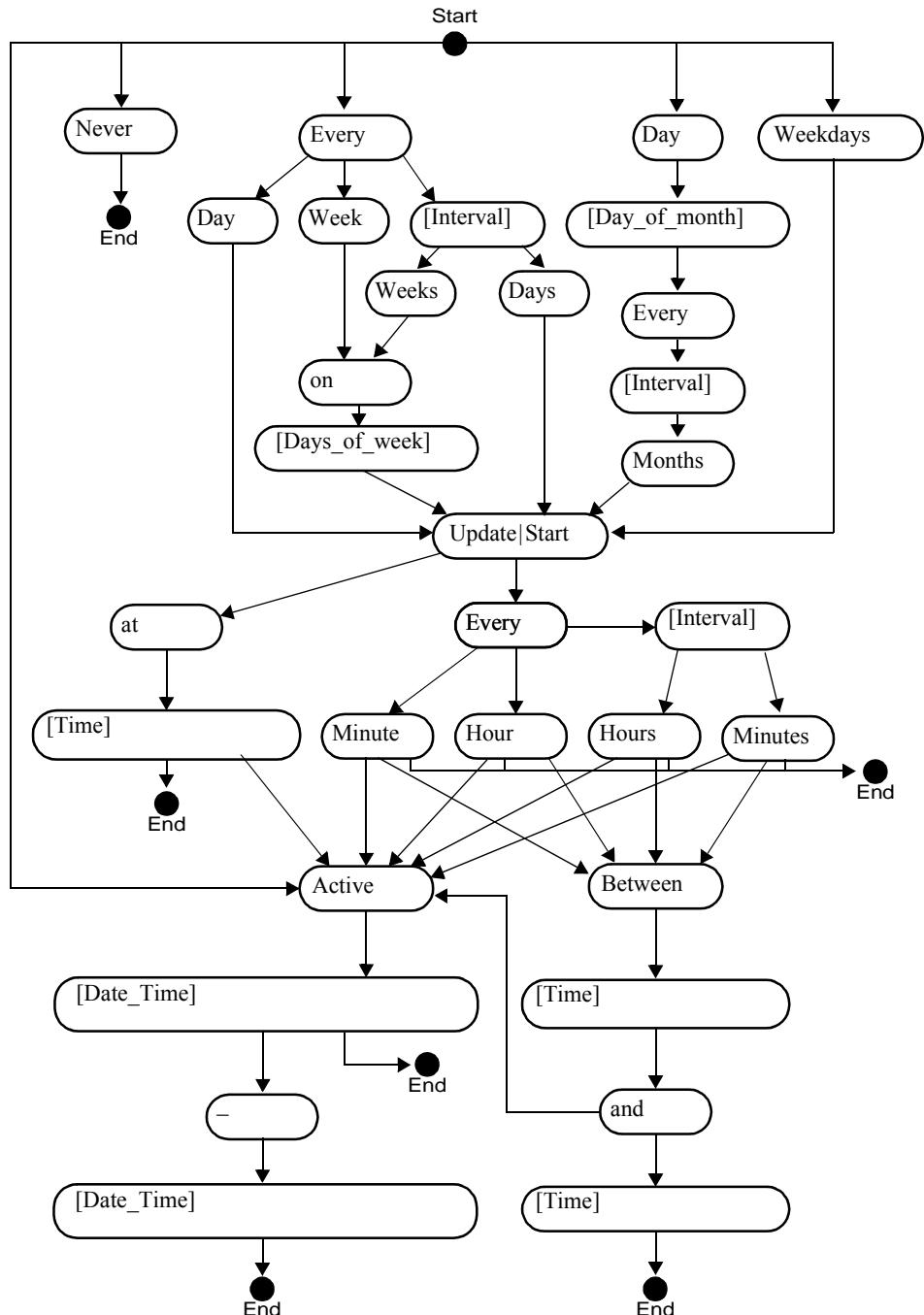
For example:

`-schedblackout "9:00AM - 5:00PM"`

Formats for primary, secondary, update, and repair schedules

The format for the other types of schedule is less straightforward and is best described in the diagram shown on the next page. A valid package schedule is any string that can be traced from the *start* state in the diagram, through intermediate states, and terminating in an *end* state.

Figure 1-1: Package schedule syntax diagram



The schedule string is not case sensitive. The following examples of package schedules are composed from the diagram:

- never
- every 2 days update at 4:00AM
- every day update every 30 minutes between 9:00AM and 5:00PM
- weekdays start every hour between 9:00AM and 5:00PM active 01/01/2002@4:00AM - 06/30/2002@6:00PM
- active 01/01/2002@4:00AM - 06/30/2002@6:00PM
- every 2 weeks on mon+wed+fri update at 4:00AM active 01/01/2002@5:00AM

Format for the primary or secondary schedule

For primary and secondary schedules, you specify a date and time when a state change occurs. Unlike in update or repair schedules, you do not need to specify a recurrence frequency. You use the following command-line options to specify a primary or secondary schedule:

```
-schedprimary {<package_URL>=<date_time_range>}  
or  
-schedsecondary {<package_URL>=<date_time_range>}
```

For example, a command that specifies a primary schedule that is active between January 1, 2002 at 4 AM and June 30, 2002 at 6 PM looks like this:

```
-schedprimary http://trans.company.com:5282/MyPackage="active 01/01/2002@4:00AM - 06/30/2002@6:00PM"
```

Format for the update or repair schedule

For update and repair schedules, you can specify a recurrence frequency. You use the following command-line options to specify an update or repair schedule:

```
-schedupdate {<package_URL>=<date_time_range_frequency>}  
or  
-schedverifyrepair {<package_URL>=<date_time_range_frequency>}
```

For example, a command to specify an update schedule that updates every two weeks on Mondays and Wednesdays at 4 AM, active beginning at 5 AM on January 1, 2002, looks like this:

```
-schedupdate http://trans.company.com:5282/MyPackage="every 2 weeks on mon+wed update at 4:00AM active 01/01/2002@5:00AM"
```

Proxy Administrator options

You can run Proxy Administrator from the command line to perform many of its functions. Use the same syntax to administer either a local or a remote proxy. The syntax for the command-line interface is as follows:

```
runchannel <ProxyAdmin_channel_URL>
[<admin_user_name>:<admin_password>@]<host>:<port> [<options>]
```

where

<ProxyAdmin_channel_URL>

specifies the channel URL for Proxy Administrator and has the form:

`http://<host>:<port>/ProxyAdministrator`

where, <host> is the host name or IP address of the transmitter from which Proxy Administrator is downloaded, and <port> is the port of this transmitter. If you do not specify a port number, 80 is used by default.

For example, if the transmitter's host name and port is `prod_Trans:80`, the Proxy Administrator channel URL can be `http://prod_Trans:80/Current/ProxyAdministrator`. Proxy Administrator is located under the /Current folder on the `prod_Trans` transmitter.

<user_name>:<password>

specifies the administration user name and password for the proxy (if configured). If the proxy uses anonymous administration, you can omit both the user name and password.

<host>

specifies the host name of the machine that hosts the proxy. If the proxy is on your local machine, you can use `localhost`.

Tip: Always provide the host name and the port of the proxy that you are administering. This practice prevents you from needing to remember exactly when to specify the host name and port and when not to (that is, when you can use `localhost`).

<port>

specifies the RPC port number of the tuner on which the proxy is running. If you do not specify a port number, the port number of the *Proxy Administrator tuner* is used.

For example, if Proxy Administrator tuner is using port 6000 as its RPC port, then this port number is used. If the proxy's tuner is actually running on port 7717, Proxy Administrator cannot connect to and administer the proxy.

Important: To administer the proxy, you must use the RPC port of the tuner hosting the proxy. This port is different from the proxy's listener port—on which the proxy listens for client requests (8080 by default).

<options>

specifies a Proxy Administrator command-line option.

In the following example, the proxy has administration restricted. Therefore, you must provide the administration user name and password with each command.

```
runchannel http://productionTx:5282/ProxyAdministrator  
user2:mypassword@proxy2:9900 -status
```

In the following example, the proxy does not have administration restricted.

```
runchannel http://productionTx:5282/ProxyAdministrator  
proxy2:9900 -status
```

The Proxy Administrator command-line options are described in the rest of this section.

-adminperms <user>:<password>

specifies the administration credentials (that is, the user name and password) for the proxy. After you set the administration user name and password, you must supply these values with all subsequent Proxy Administrator commands. By default, anyone can administer the proxy.

You must always specify the administration user name and password in plain text. The format for setting a new user name and password and for changing an existing user name and password follows:

- If you are setting the administration user name and password for a proxy that does *not already* have a user name and password configured, use the following command:

```
runchannel <ProxyAdmin_channel_URL> <host>:<port> -adminperms  
<user>:<password>
```

where *<user>* and *<password>* are the user name and password you want assigned to the proxy.

- If you want to change the administration user name and password for a proxy that *already has* an administrator user name and password configured, use the following command:

```
runchannel <ProxyAdmin_channel1_URL>  
<old_user>:<old_password>@<host>:<port> -adminperms  
<new_user>:<new_password>
```

where `<old_user>` and `<old_password>` are the current user name and password, and `<new_user>` and `<new_password>` are the new user name and password.

- If you want to allow any one to administer the proxy, use the following command:

```
runchannel <ProxyAdmin_channel1_URL>  
<old_user>:<old_password>@<host>:<port> -adminperms any  
-cachelowwm <percentage>
```

specifies a percentage that represents the lower limit (cache low watermark) for the cache size. When the proxy starts cache garbage collection, it takes a snapshot of the cache and then determines the number of files it must delete to reach the low watermark.

It is recommended that you set the cache low watermark to a value between 75 and 80 (indicating that it is 75% to 80% of the maximum cache size).

Default value: 75

```
-cachesize <size>
```

specifies the total size (in MB) for the proxy's cache. The proxy's cache does not exceed this disk-space limit. Once the cache reaches the maximum cache size, the proxy starts garbage collection to delete older channel files from the cache.

Tip: Specify a maximum cache size that is (at the most) 90% of the disk space.

Remember that the cache garbage collection process starts if the total amount of disk space on the proxy's machine is less than 10%.

Default value: 1024

-clearcache

empties the proxy's cache. You can use this option if you need a large amount of disk space for some administrative purpose. You can also use this option if you are going to back up the proxy machine without including the proxy's cache.

Once the proxy's cache is emptied, the proxy rebuilds the cache as it receives new requests. You can also pre-load the cache using -refreshcache.

Important: You must use the -stop option to stop the proxy from accepting requests before you empty the proxy's cache.

-concurrency <max connections>

specifies the number of concurrent connections to the proxy.

Tip: You should balance the number of concurrent connections with the proxy machine's hardware and network bandwidth. That is, if the amount of memory allotted to the proxy machine is not large, or if the processor and bus are slow, you should not set the number of concurrent connections to a high number.

Default value: 1024

-disableSSL

disables SSL on the proxy.

-enableSSL <SSLCertificateName> <SSLpwd> <saveSSLPwd>
 <sslClientAuth>

enables SSL on the proxy where

<SSLCertificateName> specifies the certificate name

<SSLpwd> specifies the certificate password

<saveSSLPwd> specifies whether or not to save the password. The value can be true or false. The default value is false.

<sslClientAuth> specifies the client authentication type. The value can be none, require, or request. The default value is none.

-getconfig

displays a list of all the proxy channel properties and their current settings. In most cases, the **-status** option better organizes and selectively displays information about property settings, but to view all the proxy channel properties, use the **-getconfig** option.

-logdir <directory>

specifies the directory where the proxy stores its `access` and `admin` logs.

For this option, use the following guidelines:

- Do *not* move the root directory and logs directory *inside* the tuner's workspace. Thus, if the proxy channel is deleted or the tuner is uninstalled, the proxy's configuration, cache, and logs are still retained.
- Specify the full path for the logs directory.
- The directory should be on the same machine as the proxy. That is, do not reference a different hard drive, network drive, or symbolic link.
- If the directory path contains spaces, enclose the entire path within double-quotes ("").
- If you specify a directory that does not already exist, the specified directory is created.
- Make sure the drive has enough disk space to accommodate the cache settings you specify with the **-cachesize** option.

The proxy's `access` and `admin` logs are, by default, stored under the proxy's root directory (`proxyroot`) that is located outside the tuner's workspace directory.

- On Windows, the default proxy logs directory is
`c:\winnt\marimba\proxyroot\logs`
- On UNIX, the default proxy logs directory is `$MARIMBAROOT/.marimba/proxyroot/logs`. If `$MARIMBAROOT` is not defined, it defaults to the user's login directory.

Note: If there is more than one proxy on the machine, the proxy's root directory becomes `proxyroot.x`, where `x` denotes a number that increases sequentially as proxies are installed on the machine. For example, the root directory of the first proxy on the machine is `proxyroot`, and the root directory of the second proxy is `proxyroot.1`.

-noproxychain

removes the existing proxy chain. After using this option, the proxy makes requests directly to the transmitters instead of connecting to another proxy.

Note: You should use the `-stop` option to stop the proxy from accepting requests before you remove the proxy chain. If you do not stop the proxy before changing its configuration, the proxy completes any transactions it is servicing, and then removes the proxy chain.

Default value: `false`. (There is no proxy chain.)

-normalproxy

changes either a reverse proxy or a secure reverse proxy to a normal proxy. Unlike the reverse proxy or the secure reverse proxy, the normal proxy is not bound to one target transmitter (the internal transmitter).

Note: You should use the `-stop` option to stop the proxy from accepting requests before you change it to a normal proxy. If you do not stop the proxy before changing its configuration, the proxy completes any transactions it is servicing, and then changes to a normal proxy.

After you change either a reverse proxy or a secure reverse proxy to a normal proxy, configure your endpoint tuners to use the normal proxy.

Default value: `false`. (The proxy is a normal proxy.)

-port <number>

specifies the proxy's listener port (that is, the port on which the proxy listens for requests). This port is different from the RPC port of the tuner hosting the proxy (which is `7717` by default).

Note: When you change the proxy's port number, the proxy is *automatically* stopped (from responding to client requests) and restarted.

After changing the proxy's listener port, make sure the proxy's clients now use this port when connecting to and sending requests to the proxy.

Default value: `8080`

-proxychain <host>:<port>

chains the administered proxy to another proxy. You can chain together only normal proxies. You can chain a normal proxy to a non-Marimba proxy.

Specify the host name and listener port number of the target proxy to which you want to chain your proxy.

Note: You should use the `-stop` option to stop the proxy from accepting requests before you create a proxy chain. If you do not stop the proxy before changing its configuration, the proxy completes any transactions it is servicing, and then establishes the proxy chain.

Default value: 8080

-refreshcache <URL1 URL2 ...>

preloads the proxy's cache with the specified channel, folder, channel segment, or with all the contents of the specified transmitter. The content you preload can originate from several different transmitters. You can also preload the proxy's cache with CAR files from the proxy's hard drive.

If the cache is preloaded, the proxy already has the files before a client requests them. The proxy does not have to contact the transmitter for these files. This option is useful when the proxy needs to download large files from the transmitter over a slow or busy connection.

Note: If you want to preload channels from a transmitter that has subscribe permissions configured, you must specify the subscribe credentials (that is, the subscribe user name and password) using the `refresh.credentials` proxy channel property.

When specifying channel segments, append the channel URL as follows:

?segment="*<segmentname>*"

Multiple channel URLs and CAR file paths must be separated by a space.

For information about how the proxy preloads its cache, see the *Infrastructure User Guide*, available on the Marimba Channel Store.

Example (Windows):

```
runchannel <ProxyAdmin_channel1_URL>
[<user>:<password>@]<host>:<port> -refreshcache http://
trans.myco.com/samegame http://trans.myco.com/
employeenews?segment="Windows,x86/any" file:///C:/backups/communicator.car
```

Example (UNIX):

```
runchannel <ProxyAdmin_channel1_URL>
[<user>:<password>@]<host>:<port> -refreshcache http://
trans.myco.com/samegame http://trans.myco.com/
employeenews?segment=Solaris,sparc/any" file:///user/proxy/
backups/communicator.car
```

If you want to load the contents of multiple transmitters (the `ny_server` and `ca_server` transmitters) both running on port 80, type:

```
runchannel <ProxyAdmin_channel1_URL>
[<user>:<password>@]<host>:<port> -refreshcache http://
ny_server:80 http://ca_server:80
```

If the channels that you want to preload are located on a transmitter that has subscribe permissions configured, then use the refresh.credentials proxy channel property to specify the subscribe credentials.

```
runchannel <ProxyAdmin_channel1_URL>
[<user_name>:<password>@]<host>:<port> -setProperty
refresh.credentials plain:<user>:<password>
```

where, `<user>` and `<password>` specify the subscribe user name and password in plain text.

Important: You *must* append `plain:` to the user name and password. You must provide *both* a subscribe user name and a password even when the transmitter is configured to authenticate according to only the password (that is, the transmitter allows any user who provides the correct password to subscribe to the channel). The *same* credentials (specified using the `refresh.credentials` property) are used for *all* the channels specified in the `-refreshcache` option. If different channels require different user names and passwords, preload the channels in multiple phases.

After you specify the subscribe permissions, you can use the `-refreshcache` option to preload the proxy's cache.

`-repaircache`

repairs a corrupt proxy cache storage.

Before you use this option, it is recommended that you use the `-verifyCache` option to see if your cache storage is corrupt and needs repairing.

Important: You must use the `-stop` option to stop the proxy from accepting client requests before you either verify or repair a proxy's cache.

The cache is repaired if the `-repaircache` option reports Repair: Cache Repaired or Repair: Cache OK. If the `-repaircache` option reports Repair: Cache Damaged, run the `-repaircache` option again.

-reverseproxy <target URL>

configures the proxy as a reverse proxy and specifies the target transmitter to which the proxy connects.

Specify the URL of the target transmitter (that is, the internal transmitter). Depending on how you set up DNS, you can list the transmitter either by host name only or by fully qualified host name.

For an SSL-enabled transmitter, use `https://` and specify the transmitter's fully qualified host name.

Note: The target of a reverse proxy can also be a load balancer (behind which several transmitters are located).

-rootdir <directory>

specifies the location of the proxy's root directory.

Moving the proxy's root directory also moves its cache directory, the `properties.txt` file, and so on, but *not* the logs directory. The contents of the cache directory are *not* moved. Also the old root directory and its subdirectories are not deleted. However, any *new* changes are recorded only in the *new* root directory.

For this option, use the following guidelines:

- When you change the proxy's root directory, the proxy is *automatically* stopped (from accepting client requests) and restarted.
- Do *not* move the root directory and logs directory *inside* the tuner's workspace. Thus, if the proxy channel is deleted or the tuner is uninstalled, the proxy's configuration, cache, and logs are still retained.
- Specify the full path for the root directory.
- The directory should be on the same machine as the proxy. That is, do not reference a different hard drive, network drive, or symbolic link.
- If the directory path contains spaces, enclose the entire path within double-quotes (").
- If you specify a directory that does not already exist, the specified directory is created.
- Make sure the drive has enough disk space to accommodate the cache settings you specify with the `-cachesize` option.

By default, the proxy's root directory (proxyroot) is stored next to the tuner's workspace directory.

- On Windows, the default proxy root directory is
c:\winnt\.marimba\proxyroot.
- On UNIX, the default proxy root directory is \$MARIMBAROOT/.marimba/
proxyroot. If \$MARIMBAROOT is not defined, it defaults to the user's login
directory.

Note: If there is more than one proxy on the machine, the proxy's root directory becomes proxyroot.x, where x denotes a number that increases sequentially as proxies are installed on the machine. For example, the root directory of the first proxy on the machine is proxyroot, and the root directory of the second proxy is proxyroot.1.

-secure {true | false}

(*Reverse proxies only*) specifies whether the proxy should run as a secure reverse proxy. This option only applies to reverse proxies. Use this option when you want to change a secure reverse proxy to either a reverse proxy or a normal proxy. (In both these cases, set this option to false.)

Note: You should use the -stop option to stop the proxy from accepting requests before you use the -secure option. If you do not stop the proxy first, the proxy completes any transactions it is servicing, and then changes its configuration.

Default value: false

-setproperty <key> <value>

enables you to set any proxy property that doesn't correspond to a command-line option.

Specify the proxy property name and value (separated with a space) as follows:

```
runchannel <ProxyAdmin_channel_URL>
[<user>:<password>@]<host>:<port> -setproperty
<property_name> <property_value>
```

`-setsslpw`

specifies the password for the SSL certificate specified using `-sslcert`. This option only applies to reverse proxies.

Use the `-setsslpw` option to interactively set the SSL password for your certificate. This ability is important on platforms such as UNIX, where other users can see the command you have executed in a process list. After you enter this option using the `runchannel` command, the proxy channel prompts you to enter the certificate password. After you enter the correct password, the secure reverse proxy starts.

For this option, use the following guidelines:

- You should use the `-stop` option to stop the proxy from accepting client requests before you configure the proxy as a secure reverse proxy.
- Before you specify the SSL certificate password, specify the SSL certificate ID using `-sslcert`.
- Specify the SSL certificate password in plain text.
- If you want to save the SSL certificate password for autostart capabilities (for proxies configured as NT services or as UNIX background processes), use the *proxy* channel's `-sslpw` and `-savesslpw` options.

`-sslcert <cert id>`

specifies the unique ID for the SSL certificate you want to use for your secure reverse proxy. This option only applies to reverse proxies.

For this option, use the following guidelines:

- You should use the `-stop` option to stop the proxy from accepting client requests before you configure the proxy as a secure reverse proxy.
- To find the SSL certificate's unique ID, start Certificate Manager, click SSL, select the certificate, and click View. Alternatively, you can use the certificate's distinguished name or nickname.
- After you specify the SSL certificate ID, you must use the `-setsslpw` or the `-sslpw` option to set the password for the SSL certificate.

After you have configured your proxy as a secure reverse proxy, if you later decide that you don't want the proxy to be SSL-enabled, set the `-secure` option to `false`.

`-sslpw <certpassword>`

specifies the password for the SSL certificate specified using `-sslcert`. This option only applies to reverse proxies.

Unlike the `-setsslpw` option (which also allows you to specify the SSL certificate password), the `-sslpw` option requires you to specify the SSL password with the option rather than entering it interactively. Thus, the `-sslpw` option does not offer as much security as `-setsslpw`.

For this option, use the following guidelines:

- You should use the `-stop` option to stop the proxy from accepting client requests before you configure the proxy as a secure reverse proxy.
- Before you specify the SSL certificate password, specify the SSL certificate ID using `-sslcert`.
- Specify the SSL certificate password in plain text.
- If you want to save the SSL certificate password for autostart capabilities (for proxies configured as NT services or as UNIX background processes), use the *proxy channel's* `-sslpw` and `-savesslpw` options.

`-start`

starts the proxy (to accept client requests). This option does not start the *proxy channel*.

`-status`

displays the configuration information about the proxy.

`-stop`

stops the proxy from accepting client requests. This option does not stop the *proxy channel*.

If you are planning to significantly change the proxy, you should stop the proxy from accepting requests before you make the changes.

When you stop the proxy, it completes any transactions it was servicing at the time you ran the `-stop` option. Therefore, after you run the `-stop` option, wait a minute or two before running any other configuration commands.

-verifycache

determines whether or not the proxy cache storage is corrupt. If it is corrupt, use the -repaircache option on page 111 to repair the cache storage.

Important: You must use the -stop option to stop the proxy from accepting client requests before you verify a proxy's cache.

Proxy Server options

The command-line syntax for using the Proxy Server channel is as follows:

runchannel <Proxy_Server_URL> [<options>]

where,

<options>

specifies a Proxy Server option.

The options are described in the rest of this section.

-cacheReport [<verbosity>]

creates a storage report about the proxy's cache and prints the report to the tuner's history log. (Search the history log for the following line to find the beginning of the report: Running storage report for proxy cache.) This report can be helpful for debugging purposes, especially in cases of storage corruption. Sending this report to Customer Support is a convenient alternative to sending an entire workspace.

If needed, you can open a Console Window before running this command, and then the report is also printed to the Console Window. Use the optional <verbosity> setting to adjust the amount of information in the report.

Valid values: 0, 1, and 2, with 2 being the most verbose

Default value: 1

-clearsslpw

deletes the SSL password that was previously saved on the proxy's machine. The next time the proxy starts up, if it is still configured in SSL mode, supply the SSL password. If you do not supply the SSL password, the proxy does not start.

-configure <file>

configures the proxy using the specified <config_file>, which is a text file containing a list of proxy channel properties. The configuration file is a convenient way to start the proxy channel and configure it with a set of properties.

These properties are added to the existing list of properties in the proxy's properties.txt file. If you enter a property that already exists in the properties.txt file, the new value (in the configuration file) overrides the value in the properties.txt file.

This configuration file can contain any of the proxy channel properties, which must be listed using the following form:

proxy.install.<proxy_property>=<property_value>

where proxy.install prefixes the actual proxy channel property name and value pair. (For a full list of proxy channel properties, see "Proxy properties" on page 227.) For example:

```
proxy.install.proxy.root=c:\\proxyroot  
proxy.install.server.connect.port=5000  
proxy.install.server.proxychain=true  
proxy.install.server.proxychain.nextProxy=proxy2:6000
```

For this option, use the following guidelines:

- Specify the full path to the configuration file.
- The directory that contains the file should be on the same machine as the proxy. That is, do not reference a different hard drive, network drive, or symbolic link.
- If the directory path contains spaces, enclose the entire path within double-quotes (").

-port <number>

specifies the listener port for the proxy (that is, the port on which the proxy listens for client requests). This port is *different* from the RPC port of the tuner hosting the proxy (which is 7717 by default).

When you set the port number for the proxy, the proxy uses the specified port until you change it. You don't need to specify a port number every time you start the proxy.

After changing the proxy's listener port, make sure the proxy's clients now use this port when connecting to and sending requests to the proxy.

Default value: 8080

-proxychain <host>:<port>

chains the administered proxy to another proxy. You can chain together only normal proxies. You can chain a normal proxy to a non-Marimba proxy.

Specify the host name and listener port number of the target proxy to which you want to chain your proxy.

Note: You should use the **-stop** option to stop the proxy from accepting requests before you create a proxy chain. If you do not stop the proxy before changing its configuration, the proxy completes any transactions it is servicing, and then establishes the proxy chain.

Default value: 8080

-reverseproxy <target URL>

configures the proxy as a reverse proxy and specifies the target transmitter to which the proxy connects.

Specify the URL of the target transmitter (that is, the internal transmitter). Depending on how you set up DNS, you can list the transmitter either by host name only or by fully qualified host name.

For an SSL-enabled transmitter, use `https://` and specify the transmitter's fully qualified host name.

Note: The target of a reverse proxy can also be a load balancer (behind which several transmitters—such as, the master and mirrors—are located).

By default, the proxy is not configured as a reverse proxy.

-rootdir <directory>

specifies the location of the proxy's root directory.

Moving the proxy's root directory also moves its cache directory, the `properties.txt` file, and so on, but *not* the logs directory. The contents of the cache directory are *not* moved. Also, the old root directory and its subdirectories are not deleted. However, any *new* changes are recorded only in the *new* root directory.

For this option, use the following guidelines:

- When you change the proxy's root directory, the proxy is *automatically* stopped (from accepting client requests) and restarted.
- Do *not* move the root directory and logs directory *inside* the tuner's workspace. Thus, if the proxy channel is deleted or the tuner is uninstalled, the proxy's configuration, cache, and logs are still retained.
- Specify the full path for the root directory.
- The directory should be on the same machine as the proxy. That is, do not reference a different hard drive, network drive, or symbolic link.
- If the directory path contains spaces, enclose the entire path within double-quotes ("").
- If you specify a directory that does not already exist, the specified directory is created.
- Make sure the drive has enough disk space to accommodate the cache settings you specify (using the -cachesize Proxy Administrator command-line option).

By default, the proxy's root directory (`proxyroot`) is stored next to the tuner's workspace directory.

- On Windows, the default proxy root directory is
`c:\winnt\.marimba\proxyroot`
- On UNIX, the default proxy root directory is `$MARIMBAROOT/.marimba/proxyroot`. If `$MARIMBAROOT` is not defined, it defaults to the user's login directory.

Note: If there is more than one proxy on the machine, the proxy's root directory becomes `proxyroot.x`, where `x` denotes a number that increases sequentially as proxies are installed on the machine. For example, the root directory of the first proxy on the machine is `proxyroot`, and the root directory of the second proxy is `proxyroot.1`.

`-savesslpw`

saves the password for the SSL certificate specified using the `-sslcert` Proxy Administrator command-line option. This option only applies to reverse proxies.

Saving the SSL password is useful for autostart capabilities and is needed by a proxy that is running as either an NT service or as a UNIX background process. You must use this option in conjunction with the `-ssplw` option to save the password for autostart capabilities.

For this option, use the following guidelines:

- Before you specify the SSL certificate password, you specify the SSL certificate ID using the `-sslcert` Proxy Administrator command-line option.
- Use the `-savesslpw` option with the `-sslpw` option. The order in which you specify these options is not important. For example:

```
runchannel <Proxy_channel_URL> -sslpw <password> -savesslpw
```

If you want to delete the SSL password, use the `-clearsslpw` option.

```
-setsslpw
```

specifies the password for the SSL certificate specified using the `-sslcert` Proxy Administrator command-line option. This option only applies to reverse proxies.

Use the `-setsslpw` option to interactively set the SSL password for your certificate. This ability is important on platforms such as UNIX, where other users can see the command you executed in a process list. After you enter this option using the `runchannel` command, the proxy channel prompts you to enter the certificate password. After you enter the correct password, the secure reverse proxy starts.

Important: If the reverse proxy is running from a tuner that is an NT service or that starts from a UNIX `init` script, you cannot use this option (because you are required to enter the password interactively). Use the `-ssplw` option instead. This option is less secure.

For this option, use the following guidelines:

- Before you specify the SSL certificate password, specify the SSL certificate ID using the `-sslcert` Proxy Administrator command-line option.
- Specify the SSL certificate password in plain text.

- If you want to save the SSL certificate password for autostart capabilities (for proxies configured as NT services or as UNIX background processes), use the `-sslpw` option with the `-savesslpw` option.

`-sslpw <certpassword>`

specifies the password for the SSL certificate that was specified using the `sslcert` Proxy Administrator command-line option. This option only applies to reverse proxies.

Unlike the `-setsslpw` option (which also allows you to specify the SSL certificate password), the `-sslpw` option requires you to specify the SSL password with the option rather than entering it interactively. Thus, the `-sslpw` option does not offer as much security as `-setsslpw`.

For this option, use the following guidelines:

- Before you specify the SSL certificate password, specify the SSL certificate ID using the `sslcert` Proxy Administrator command-line option.
- Specify the SSL certificate password in plain text.
- If you want to save the SSL certificate password for autostart capabilities (for proxies configured as NT services or as UNIX background processes), save the SSL certificate password using the `-sslpw` option *with* the `-savesslpw` option. The order in which you specify these options is not important.

For more information about saving the SSL certificate password for autostart capabilities, see the *Proxy User Guide*, available on the Marimba Channel Store.

Publisher options

You can run Publisher from the command line to perform many of its functions. The syntax for the command-line interface is as follows:

```
runchannel <Publisher_URL>
-d <channel_directory> -url <channel_URL>
[<options>]
```

where

`-d <channel_directory>`

specifies the channel directory on your local file system (for example, C:\MyChannel on a Windows system). The contents of this directory are published to the channel URL specified in the -url option.

When a channel is republished, the contents of the channel directory are compared to the contents of already published channel, and the latter is updated accordingly. Thus, if the published channel contains files that are not in the channel directory, those files are deleted.

-url <channel_URL>

specifies the destination channel URL on the transmitter — the one to which the command publishes.

<options>

specifies the Publisher command-line option.

The Publisher command-line options are described in the rest of this section.

-add <target_path> <source_path>

adds the specified file or directory to the published channel (circumventing the channel directory).

<target_path>

is a path, relative to the URL specified in the -url option, specifying where the file or directory is to be published on the transmitter.

<source_path>

is an absolute path specifying the local file or directory to add.

Because -add and -addf (below) circumvent the channel directory, these options are one-time operations. If you republish the channel, you must again explicitly add that file (either with -add or -addf again or in the channel directory itself), or the file is deleted from the published channel. For example, if you add a file when you first publish a channel as follows, my.txt is added to the published channel on the transmitter but not to the channel directory:

-add my.txt C:\temp\my.txt

If you publish the channel again as follows, or with no -add option at all (and you haven't explicitly stored my.txt in the channel directory), my.txt is deleted from the published channel when its contents are compared to the contents of the channel directory:

-add your.txt C:\temp\your.txt

-addf <file>

adds the files listed in the specified additions file to the published channel (circumventing the channel directory). The additions file is a text file in which each line has the form

<target_URL>, <source_path>

For more information about this option, refer -add.

-capabilities all | none

specifies the capabilities the channel needs when run on the tuner: all means the channel needs at least one type of special capability, such as printer, network, or file access; none means the channel doesn't need any special access to the user's computer.

-clientcertpw <certificate_password>

sends the client certificate password to the tuner. Use this option if the transmitter requires a client certificate.

-help

lists and briefly describes all the Publisher command-line options.

-ignore <filter>

specifies the base directory exclusion filter. You can use wildcard characters for files or directories that you want to exclude from the channel.

-output <file>

specifies a file in which Publisher stores its output.

-password <password>

specifies the password for the publish operation. The password can be either a user password (for version 4.x transmitters) or a transmitter/channel password (for 3.x and older transmitters). Depending on the transmitter settings, you might have to use this option in conjunction with the -user option.

-preview

previews the publish operation but doesn't actually publish the channel.

```
-propsfile <file>
```

copies the specified file to the channel's properties.txt file. This option lets you create a specific properties.txt file for the channel rather than use the default file.

```
-q
```

suppresses output during publishing. Publisher does not display any status or confirmation messages at the command line. (q stands for "quiet.")

```
-segment <segment_ID>
```

specifies a channel segment to publish.

Note: If you are publishing . segments (for example, .authenticator), you shouldn't have platform and locale properties set in your properties.txt file. If you do, you'll need to use the -segment option to override the platform and locale properties.

```
-sign <cert_name> <cert_password> all | signed
```

signs the channel with the specified certificate and capabilities.

```
-signfile <file>
```

signs the channel with the certificate and capabilities specified in <file>.

```
-unpublish
```

unpublishes the channel.

```
-user <user_name>
```

sets a user name for publishing to transmitters that have access control. This option is used in conjunction with the -password option.

```
-v
```

enables verbose output during publishing. Publisher displays status and confirmation messages at the command line.

For example:

```
runchannel http://trans/Publisher -d C:\MyChannel -url http://  
jim:80/MyChannel  
-add my.txt C:\temp\my.txt
```

publishes the contents of the channel directory C:\MyChannel to http://jim:80/MyChannel and adds the file C:\temp\my.txt to the published channel.

```
runchannel http://trans/Publisher
-d C:\MyChannel -url http://jim:80/MyChannel
-add newstuff/my.txt C:\temp\my.txt
```

publishes the contents of the channel directory C:\MyChannel to http://jim:80/MyChannel and adds the file C:\temp\my.txt in a subdirectory of the published newstuff channel.

Report Center options

You can run Report Center from the command line to configure the Inventory and Logging plug-ins and to perform some other tasks, such as running a collection or exporting queries to an XML file. The syntax of the command-line interface is as follows:

```
runchannel <ReportCenter_URL> -user <user_name> -password
<password> <options>
```

where <ReportCenter_URL> is the URL of the Report Center channel, and <user_name> and <password> are the user name and password that you use to log in to Report Center. The Report Center options are described in the rest of this section.

If you want to use the command-line interface to configure any plug-in settings, you must use a number of options in combination. These required plug-in options are shown in the following example (for your convenience, each option is shown on a separate line, although in practice, you enter all the options without any break until the end of the command):

```
runchannel http://trans.acme.com:5282/ReportCenter
-user primary
-password mypassword
-url http://inventory.acme.com:5282/InventoryService
-serviceType inventory
-dbType oracle
-dbHost inventory.marimba.com
-dbPort 1521
-dbName indy
-dbUser inventory
-dbPw inventory
```

You must enter all these options, along with any other plug-in or endpoint settings you want to use to identify the plug-in you are configuring.

Note: Although the capitalization of the following option names is shown in mixed-case letters, you can enter the option names using all lowercase letters if that is more convenient. For example, instead of typing `-pluginLogRollPolicy`, you can type `-pluginlogrollpolicy`. The capitalization is shown in mixed case so that the option names are easier to read.

`-appExtUrl {<channel_url> | none}`

(optional) enables or disables a scanner extension channel that runs at the time when the application-specific scan is done, as follows: Use `<channel_url>` if you want to use a scanner extension channel and specify its URL, or use `none` if you want to turn the scanner extension channel off. Only primary administrators are allowed to set this option.

`-appScanSchedule <schedule_string>`

(optional) specifies the schedule for the application-specific scan. For details and examples, see “Syntax for the schedule string” on page 259.

`-cancelButton {true | false}`

(optional user interaction property) specifies, if set to `true`, that a Cancel button appears in the progress window that users see when an inventory scan is performed on their machines. Clicking this Cancel button enables users to cancel both the progress window and the scan. Only primary administrators are allowed to set this option.

Default value: `false`

`-dbHost <host_name>`

(required when using any plug-in configuration options) specifies the database host to which you want to send inventory scans and filtered log files, where `<host_name>` is the name of the computer running the database. Only primary administrators are allowed to set this option.

`-dbName <database_name>`

(required when using any plug-in configuration options) specifies the system ID (for Oracle databases) or the database name (for Microsoft SQL Server 2000 databases), where `<database_name>` is the database name or SID. Only primary administrators are allowed to set this option.

Default value: `invdb` (For SQL Server, the default database name is `invdb`, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)

```
-dbPort <port>
```

(required when using any plug-in configuration options) specifies the server port number for accessing the database, where `<port>` is the port number. The commonly used port numbers are 1521 for Oracle and 1433 for Microsoft SQL Server 2000. Only primary administrators are allowed to set this option.

```
-dbPw <database_password>
```

(required when using any plug-in configuration options) specifies the password for accessing the database, where `<database_password>` is the password. Use `inventory` for the password unless you edited the database installation scripts to change this value. Only primary administrators are allowed to set this option.

Note: The default password, `inventory`, is used with this option so that you can *insert data into* the database. When you set up a data source (in the Applications > Console > System Settings menu, Data Source tab, Database link) to specify which database you want to query, that is, *get data out of* the database, the default you use is `user_view`.

```
-dbType {oracle | sqlserver2000}
```

(required when using any plug-in configuration options) specifies whether you are using an Oracle database or a Microsoft SQL Server 2000 database for storing your inventory scan reports and filtered log files. Only primary administrators are allowed to set this option.

```
-dbUser <user_name>
```

(required when using any plug-in configuration options) specifies the user name for accessing the database, where `<user_name>` is the user name. Use `inventory` for the user name unless you edited the database installation scripts to change this value. Only primary administrators are allowed to set this option.

```
-deletePluginFromDb
```

(optional) deletes entries from the Select Plug-in list on the Plug-in to Configure page of the Report Center GUI. Use this option if you accidentally enter an invalid URL when trying to connect to the Inventory or Logging plug-in, and you don't want this invalid URL to appear in the drop-down list. Use this option in conjunction with the `-serviceType` option, to specify which type of plug-in (inventory or logging), and use the `-url` option to specify which URL you want to remove from the list.

Example: (Although for display purposes, the options are shown on several lines, in actual practice, you enter all the options without any break until the end of the command.)

```
runchannel http://trans.acme.com:5282/ReportCenter  
-user primary -password mypassword  
-deletePluginFromDb  
-url http://inventy.acme.com:5282/InventoryService  
-serviceType inventory  
-exportQueries <query_path> <file>
```

(optional) enables you to export queries to an XML file, where `<file>` is the complete path to the file, including drive letter on Windows. You can then import these queries into another Report Center. You have the option of exporting one query, one folder of queries, or all queries. Only primary administrators and standard administrators with unrestricted access are allowed to set this option.

Example of exporting one query (called `Query1` in the folder `Folder1`):

```
-exportQueries /Folder1/Query1 c:\tmp\query1.xml
```

UNIX example:

```
-exportQueries /Folder1/MyQuery /home/Folder1/queries/  
myquery.xml
```

Example of exporting a folder:

```
-exportQueries /Folder1 c:\tmp\folder1.xml
```

Example of exporting all queries:

```
-exportQueries / c:\tmp\all.xml
```

Note: When specifying the query path, use forward slashes as path separators.

```
-h or -help [<option>]
```

(optional) lists and briefly describes all the Report Center command-line options. To display help about a specific command-line option, use `-help <option>`, where `<option>` is the name of the option. Do not use a hyphen with `<option>` (for example, to get information about the `-importQueries` option, you type `-help importQueries`).

`-importCollections [-dbType {oracle | sqlserver2000}]`

(optional) migrates Subscription 4.7 collections queries to Report Center. The process is as follows: First, make sure that you configure Report Center to use the LDAP server that contains your 4.7 collections queries. Next use the Report Center `-importCollections` option. The 4.7 collections are exported, the syntax of each query is checked to make sure it specifies `distinct`, and then the collections queries are automatically imported into the database that Report Center uses. Only primary administrators are allowed to set this option.

If the SQL syntax of the query does not specify `distinct`, the collection is exported, but a message warns you that you should correct the syntax after the query is imported into Report Center. The collections are exported to a file, called `collections4x.xml`, so that you have a record of your 4.7 collections. You can also use this file if you want to import these collections into another Report Center.

If you omit the `-dbType` option, Report Center assumes that the SQL syntax for the query is suitable for the database type that Report Center uses. In most cases, the `-dbType` option is not necessary. If you do use the `-dbType` option, Report Center validates that the syntax works for that specific type of database. You can specify `oracle` (for Oracle databases) or `sqlserver2000` (for Microsoft SQL Server 2000). If the syntax is not correct for the database type, Report Center does not import the queries, and displays an error message.

Note: After you use the `-importCollections` option, make sure that the syntax of your queries contains the `machine_name` column. Run the queries, either manually or by setting a schedule. The process of running the queries actually upgrades them in LDAP.

-importQueries <file>

(optional) enables you to import queries that were previously exported from Report Center, where <file> is the path to the XML file that contains the queries. The path can be relative or absolute, for example: -importQueries ..\..\xml\myfile.xml (relative) or -importQueries c:\xml\myfile.xml (absolute). Only primary administrators and standard administrators with unrestricted access are allowed to set this option.

If any of the imported queries or folders have the same name as existing queries or folders, the imported names overwrite the existing names. If the queries are exported from an Oracle database, you can get a warning message if you import them into an SQL Server database, and vice versa.

Because this is not a plug-in configuration option, you do not need to use any of the -db* options or the -url option with this option.

-infraExtUrl {<channel_url> | none}

(optional) enables or disables a scanner extension channel that runs at the time when the infrastructure-specific scan is done, as follows: Use <channel_url> if you want to use a scanner extension channel and specify its URL, or use none if you want to turn the scanner extension channel off. Only primary administrators are allowed to set this option.

-infraScanSchedule <schedule_string>

(optional) specifies the schedule for the infrastructure-specific scan. For details and examples, see “Syntax for the schedule string” on page 259.

-maxDbConn <number>

(optional) specifies the maximum number of simultaneous connections between the plug-in and database. Only primary administrators are allowed to set this option.

Default value: 5

-minDbConn <number>

(optional) specifies the minimum number of simultaneous connections between the plug-in and database. Only primary administrators are allowed to set this option.

Default value: 2

-password <password>

(required with all other options; however, if you have not set a password, you do not need to use this option) specifies the password you use to log in to Report Center. This option, in combination with the -user option, authenticates you to Report Center and enables Report Center to determine whether you have primary administrator, standard administrator, or operator privileges.

-pluginLogRollPolicy {hourly | daily | weekly | monthly | yearly | never | bysize}

(optional) specifies when to roll the log files. Only primary administrators are allowed to set this option.

Default value: daily

-pluginLogRollSize <number_in_KB>

(optional) specifies, if the roll policy is set to bysize, how large the log size can get, in KB, before the log is rolled. Only primary administrators are allowed to set this option.

Default value: 500

-pluginLogRollVersions <number>

(optional) specifies the number of log files to retain, in addition to the current log file. Only primary administrators are allowed to set this option.

Default value: 14

-pluginMaxDiskSize <number_in_MB>

(for the Logging plug-in only) specifies the maximum disk space allowed for rolled compressed log files in the transmitter's logging queue directory, in MB. When accumulated log data exceeds this size, additional data from endpoints is refused until the plug-in inserts data into the database and deletes the data from its queue. Only primary administrators are allowed to set this option.

Default value: 100

-pluginMaxFiles <number>

(for the Logging plug-in only) specifies the maximum number of rolled compressed log files allowed in the transmitter's logging queue, where <number> is the number of files. Each file requires a certain amount of memory. Therefore, if you see out-of-memory messages, you might need to adjust this setting. The default setting is based on the assumption that you are using the recommended heap sizes for the memory allocation pool (128 MB minimum and 256 MB maximum). Only primary administrators are allowed to set this option.

Note: Report Center observes both this setting and the setting for maximum plug-in disk size. Therefore, if the maximum number of files is set to 500,000 and the maximum disk size is set to 100 MB, then if 200,000 files take up 100 MB, the queue is not able to hold more than the 200,000 files.

Default value: 500000

-pluginState {on | off}

(optional and for the Inventory plug-in only) sets the Inventory plug-in to be enabled (on) or disabled (off) for collecting scan reports from endpoints. Before you update the Scanner Service channel or upgrade your database, it is recommended that you disable the plug-in. (This feature is not available for a plug-in whose version is earlier than 5.0.2.) Only primary administrators are allowed to set this option.

Default value: on

-progressBar {true | false}

(optional user interaction property) specifies, if set to true, that a progress window is displayed to the user during an inventory scan. Only primary administrators are allowed to set this option.

Default value: true

-prompt {true | false}

(optional user interaction property) specifies, if set to true, that the following prompt is displayed before beginning the first inventory scan: The Inventory scanner would like to scan your machine. Do you agree? The user can answer yes or no, and then select whether the prompt is allowed to appear again. Only primary administrators are allowed to set this option.

Default value: true

-pubPw <publish_password>

(optional) specifies the password required if you have publish permissions set on the transmitter. Use this option in conjunction with the -pubUser option, which specifies the user name associated with the password. Only primary administrators are allowed to set this option.

-pubUser <publish_user_name>

(optional) specifies the user name required if you have publish permissions set on the transmitter. Use this option in conjunction with the -pubPw option, which specifies the password associated with the user name. Only primary administrators are allowed to set this option.

-range <range_of_IDs>

(optional) specifies for Logging Service a range of channel log IDs so that endpoint tuners forward only the logging data with the IDs in the specified range (for example, 1000-5000). Only primary administrators are allowed to set this option.

If you want to specify more than one range, use a comma-separated list (for example, 1000-5000,26000-26999). Do not use spaces or carriage returns between the range values.

For a list of the ID ranges for various product channels, see “Log ID ranges for specific channels” on page 311. For a list of individual log IDs and messages for each channel, see the “*Logging codes*” on page 307.

-repeaterInsert {true | false}

(optional) specifies, if set to true, that repeaters are to send scan reports or filtered logs directly to the database. Only primary administrators are allowed to set this option.

Default value: false

-runCollection "<query_name>"

(optional) runs a collection, where <query_name> is the name of the query you want to run. Use double quotation marks around the query name if the name contains spaces. This option is useful if you want to use scripting to automate some Report Center processes. Only primary administrators and standard administrators are allowed to run collections to which they have access.

Because this is not a plug-in configuration option, you do not need to use any of the -db* options or the -url option with this option.

```
-scan {all | none | tuner+application+system+wmi}
```

(optional) sets the basic type of inventory information that is scanned for:

- Use `all` if you want all the components to be scanned for.
- Use `none` if you don't want any of the components to be scanned for. There is no equivalent GUI option for the `none` command-line argument. If you use `none` and then later decide that you want to scan for some components, change the setting by using the command line. The GUI options do not work as long as the argument is set to `none` at the command line.
- Use some combination of `tuner+application+system+wmi` to specify one to three of the possible information types to scan for. To scan for more than one type, separate your entries with a plus sign (+). If you want to scan for all four types, use the `all` argument.

Only primary administrators are allowed to set this option.

Note: If you don't have the Inventory Management module, that is, if you have only the basic infrastructure version of the inventory management system, then the only scan options you can use are `tuner` and `none`.

```
-scannerState {on | off}
```

(optional) turns inventory scanner services on or off. If you set this option to `off`, the Scanner Service still checks for updates according to schedule, but it does not perform a scan. Only primary administrators are allowed to set this option.

```
-schedule <schedule_string>
```

(optional) specifies the update schedule for the Logging Service. For details and examples, see “Syntax for the schedule string” on page 259.

Note: To set schedules for inventory scans, use the options

`-appScanSchedule`, `-infraScanSchedule`, and `-sysScanSchedule`.

```
-serviceType {logging | inventory}
```

(required when using any plug-in or endpoint configuration options) specifies the plug-in type. Use `logging` when configuring the Logging plug-in, and use `inventory` when configuring the Inventory plug-in. Only primary administrators are allowed to set this option.

-setAcl {on | off}

(optional) enables or disables access control lists (ACLs) for Report Center. Before enabling ACLs, use the console to make sure that users are authenticated using the directory service for Report Center and that data from the directory service is synchronized with the database. Then set up access control lists in the console. Only primary administrators are allowed to set this option.

-severity {CRITICAL | MAJOR | MINOR | WARNING | CLEARED | INFO | AUDIT}

(optional) specifies for the Logging Service that endpoint tuners are to forward logging data with this minimum level of severity to the transmitter. The severity levels are (from most severe to least severe) CRITICAL, MAJOR, MINOR, WARNING, CLEARED, INFO, AUDIT. Each level includes the messages from all the levels below it. Only primary administrators are allowed to set this option.

For detailed descriptions of the severity levels, see “Log severity levels” on page 309. For a list of individual log IDs and severity levels for each channel, see the “*Logging codes*” on page 307.

-softwareUsage {true | false}

(optional) specifies whether you want to capture software usage information on the endpoints. You must have the Software Usage component to use this option. Only primary administrators are allowed to set this option.

-status

(optional) displays the current configuration of the specified plug-in. This option is the command-line equivalent of the Preview Configuration Changes page in the Report Center browser-based interface. Only primary administrators are allowed to use this option.

-subPw <subscribe_password>

(optional) specifies the password required if you have subscribe permissions set on the transmitter. Use this option in conjunction with the -subUser option, which specifies the user name associated with the password. Only primary administrators are allowed to set this option.

-subUser <subscribe_user_name>

(optional) specifies the user name required if you have subscribe permissions set on the transmitter. Use this option in conjunction with the -subPw option, which specifies the password associated with the user name. Only primary administrators are allowed to set this option.

-sysExtUrl {<channel_url> | none}

(optional) enables or disables a scanner extension channel that runs at the time when the system/hardware-specific scan is done, as follows: Use <channel_url> if you want to use a scanner extension channel and specify its URL, or use none if you want to turn the scanner extension channel off. Only primary administrators are allowed to set this option.

-sysScanSchedule <schedule_string>

(optional) specifies the schedule for the system/hardware-specific scan. For details and examples, see “Syntax for the schedule string” on page 259.

-triggerOn <list_of_IDs>

(optional) specifies for Logging Service a comma-separated list of channel log IDs so that when an event with the specified ID occurs, the endpoint tuner rolls all log files and forwards them to the plug-in on the transmitter. Only primary administrators are allowed to set this option.

The triggering log IDs need to be within the range you specify by using the -range option and also be of at least as high a severity level as that which you specify by using the -severity option. For example, if the severity level for filtering is set to MAJOR, but the log ID you want to use for a trigger has a severity level of INFO, you receive an error.

For a list of the ID ranges for various product channels, see “Log ID ranges for specific channels” on page 311. For a list of individual log IDs and messages for each channel, see the “*Logging codes*” on page 307.

-url <Service_URL>

(required when using any plug-in or endpoint configuration options) specifies the URL of the Scanner Service channel or the Logging Service channel you want to configure. Only primary administrators are allowed to set this option.

```
-user <user_name>
```

(required with all other options) specifies the user name you use to log in to Report Center. This option, in combination with the `-password` option, authenticates you to Report Center and enables Report Center to determine whether you have primary administrator, standard administrator, or operator privileges.

The following example illustrates a typical combination of options you can use to configure an Inventory plug-in. (Although for display purposes, the options are shown on several lines, in actual practice, you enter all the options without any break until the end of the command.)

```
runchannel http://trans.acme.com:5282/ReportCenter
-user primary -password mypassword
-url http://inventory.acme.com:5282/InventoryService
-serviceType inventory
-appScanSchedule weekdays update every 12 hours
-sysScanSchedule every 1 days update at 12:00am
-infraScanSchedule every 1 weeks on mon update at 04:00am
-scannerState on
-dbType oracle -dbHost inventory.marimba.com -dbPort 1521 -dbName
indy -dbUser inventory -dbPw inventory
```

Schema Manager options

You can use Schema Manager command-line options to perform many of its functions.

The syntax of the command-line interface is as follows:

```
runchannel <SchemaManager_URL> -user <user_name> -password
<password> <options>
```

where

- `SchemaManager_URL`: URL of the Schema Manager channel
- `user_name` and `password`: credentials that you use to log in to Schema Manager
- `<options>` contains one of the following Schema Manager command-line arguments and attributes:

```
-export_dbtree -dbtype <databaseType> -host <hostIpAddress>
-port <port> -sid <instanceName> -sysuser <systemUserName>
-syspassword <systemPassword> -inventory_password
<inventoryPassword> -user_view_password <user_viewPassword>
[-datafiledirectory <pathToExportDirectory>] [-version latest]
```

instructs Schema Manager to export dbtree information from the database to an xml file.

- *databaseType*: sqlserver or oracle
- *hostIpAddress*: host name or IP address of the database server
- *port*: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- *instanceName*: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- *systemUserName*: is sa
- *systemPassword*: sa
- *inventoryPassword*: inventory (default password)
- *user_viewPassword*: user_view (default password)
- *pathToExportDirectory* (optional): path to the location of the files to write the files
- latest (optional): if you do not specify the -version argument and latest, the export utility exports version 1.0.

```
-generate_ldap_schema <-basedn "dc=example,dc=com" > <-d  
"ldif_file_location">
```

This parameter instructs Schema Manager to generate the configuration file that contains the ADAM / AD LDS schema.

```
-import_dbtree -dbtype <databaseType> -host <hostIpAddress>  
-port <port> -sid <instanceName> -sysuser <systemUserName>  
-syspassword <systemPassword> -inventory_password  
<inventoryPassword> -user_view_password <user_viewPassword>  
[-datafiledirectory pathToExportDirectory] [-forceimport true]
```

instructs Schema Manager to import the dbtree information from the xml file back to the database.

- *databaseType*: sqlserver or oracle
- *hostIpAddress*: host name or IP address of the database server
- *port*: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- *instanceName*: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- *systemUserName*: is sa

- *systemPassword*: sa
- *inventoryPassword*: inventory (default password)
- *user_viewPassword*: user_view (default password)
- *pathToExportDirectory* (optional): path to the location of the files to import
- true (optional): unless you add the -forceimport argument and true, the import utility does not import data values

The `export_dbtree` and `import_dbtree` commands enable you to implement the updated table and data structure for the schema. This updated structure consists of newly formatted unique code and data types. You use these commands only when you perform a manual schema update. You must first export the dbtree information from the database into an XML file and then import the XML file back to the database. You should not stop Schema Manager between commands.

```
-install_db_schema -dbtype <databaseType> -host <hostIpAddress>
-port <port> -sid <instanceName> -sysuser <systemUserName>
-syspassword <systemPassword> <installCredentials> -schemas
"core,patch,swusage,pda,swcompliance,health_monitoring" -datafile
"path_to_MSDE_data_directory" [-size "200"]


- databaseType: sqlserver or oracle
- port: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- instanceName: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- systemUserName: is sa
- systemPassword: sa
- inventoryPassword: inventory (default password)
- user_viewPassword: user_view (default password)
- installCredentials for Inventory database schema: -inventory_password inventory -user_view_password user_view
- installCredentials for Infrastructure Status Monitor database schema: -hmadmin_password hmadmin -hmuser_password hmuser

```

This argument instructs Schema Manager to install the specified Marimba schemas.

```
-reinstall_db_schema -dbtype <databaseType> -host <hostIpAddress>
-port <port> -sid <instanceName> -sysuser <systemUserName>
-syspassword <systemPassword> <installCredentials> -schemas
"core,patch,swusage,pda,swcompliance,health_monitoring" <-datafile
"path_to_MSDE_data_directory" [-size "200"]
```

- *databaseType*: sqlserver or oracle
- *port*: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- *instanceName*: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- *systemUserName*: is sa
- *systemPassword*: sa
- *inventoryPassword*: inventory (default password)
- *user_viewPassword*: user_view (default password)
- *installCredentials* for Inventory database schema: -inventory_password inventory -user_view_password user_view
- *installCredentials* for Infrastructure Status Monitor database schema: -hmadmin_password hmadmin -hmuser_password hmuser

This parameter instructs Schema Manager to reinstall the specified Marimba schemas.

```
-uninstall_db_schema -dbtype <databaseType> -host <hostIpAddress>
-port <port> -sid <instanceName> -sysuser <systemUserName>
-syspassword <systemPassword> <installCredentials> -schemas
"core,patch,swusage,pda,swcompliance,health_monitoring"
```

- *databaseType*: sqlserver or oracle
- *port*: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- *instanceName*: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- *systemUserName*: is sa
- *systemPassword*: sa
- *inventoryPassword*: inventory (default password)
- *user_viewPassword*: user_view (default password)

- *installCredentials* for Inventory database schema: -inventory_password inventory -user_view_password user_view
- *installCredentials* for Infrastructure Status Monitor database schema: -hmadmin_password hmadmin -hmuser_password hmuser

This parameter instructs Schema Manager to uninstall the specified Marimba schemas.

```
-upgrade_db_schema -dbtype <databaseType> -host <hostIpAddress>
-port <port> -sid <instanceName> -sysuser <systemUserName>
-syspassword <systemPassword> <installCredentials> -schemas
"core,patch,swusage,pda,swcompliance,health_monitoring"
```

- *databaseType*: sqlserver or oracle
- *port*: 1521 (Oracle) or 1443 (Microsoft SQL Server)
- *instanceName*: invdb (For SQL Server, the default database name is invdb, unless you edited the necessary database setup batch scripts to change this value during the install, reinstall, or upgrade process.)
- *systemUserName*: sa
- *systemPassword*: sa
- *inventoryPassword*: inventory (default password)
- *user_viewPassword*: user_view (default password)
- *installCredentials* for Inventory database schema: -inventory_password inventory -user_view_password user_view
- *installCredentials* for Infrastructure Status Monitor database schema: -hmadmin_password hmadmin -hmuser_password hmuser

This parameter instructs Schema Manager to upgrade the specified Marimba schemas.

Transmitter options

You can run a transmitter from the command line to perform many of its functions. The syntax for the command-line interface is as follows:

```
runchannel <transmitter_URL> [<options>]
```

where

<options>

specifies a Transmitter command-line option.

The Transmitter command-line options are described in the rest of this section.

-clearsslpw

deletes the SSL password that was previously saved on the transmitter's machine. The next time the transmitter starts, if it is still configured in SSL mode, supply the SSL password. If you don't supply the SSL password, the transmitter does not start.

-disableSSL

disables SSL on the transmitter.

-enableSSL <SSLCertificateName> <SSLpwd> <saveSSLPwd>
<sslClientAuth>

enables SSL on the transmitter where

<SSLCertificateName> specifies the certificate name

<SSLpwd> specifies the certificate password

<saveSSLPwd> specifies whether or not to save the password. The value can be true or false. The default value is false.

<sslClientAuth> specifies the client authentication type. The value can be none, require, or request. The default value is none.

-f <config_file>

configures the transmitter using settings in *config_file*. The configuration file is a text file that should contain properties in the form:

tx.install.<transmitter_property>=<property_value>

where tx.install prefixes the actual transmitter property name and value pair. For example:

```
tx.install.main.transmitter.root=c:\\transroot  
tx.install.transmitter.http.port=5000
```

-help

-h

lists and briefly describes all the transmitter command-line options.

-savesslpw

saves the SSL password for future use. This password is automatically used when the transmitter starts; you are not prompted to supply the SSL password.

-setsslpw

specifies an SSL password that is used to initialize SSL when the transmitter starts. If the transmitter isn't already running, this option starts the transmitter in SSL mode. You are prompted for a password at the command line. Before using this option, you start the transmitter in SSL mode from the user interface. When you start SSL, the `transmitter.http.dn` property is set in the `properties.txt` file for the transmitter.

You can use the following command-line options when starting the transmitter with `tuner -start`:

-bindaddr <*bind_address*>

uses a specific *bind_address* when listening on the main listener port.

-fileStorageReport [<*verbosity*>]

creates a report about the transmitter's storage files and prints the report to the tuner's history log. (Search the history log for the following line to find the beginning of the report: `Running storage report for file storage.`) This report can be helpful for debugging purposes, especially in cases of storage corruption. Sending this report to Customer Support is a convenient alternative to sending an entire workspace.

If needed, you can open a Console Window before running this command, and then the report is also printed to the Console Window. Use the optional <*verbosity*> setting to adjust the amount of information in the report.

Valid values: 0, 1, and 2, with 2 being the most verbose

Default value: 1 (This report can be quite long, and it is therefore recommended that you start with the default setting of 1, rather than 2.)

-fsck

checks the tuner's workspace directory for corruption and attempts to fix errors found in the tuner's workspace directory.

Note: Make sure you stop the tuner before using the `-fsck` option. Do not use the file system check option when the tuner is running.

-httpport <port>

uses <port> as the main listener port for the transmitter.

-indexStorageReport [<verbosity>]

creates a report about the transmitter's index storage and prints the report to the tuner's history log. (Search the history log for the following line to find the beginning of the report: Running storage report for index storage.) This report can be helpful for debugging purposes, especially in cases of storage corruption. Sending this report to Customer Support is a convenient alternative to sending an entire workspace.

If needed, you can open a Console Window before running this command, and then the report is also printed to the Console Window. Use the optional <verbosity> setting to adjust the amount of information in the report.

Valid values: 0, 1, and 2, with 2 being the most verbose

Default value: 1

-rootdir <directory>

sets the transmitter's workspace directory (by setting the value of the transmitter's `main.transmitter.root` property), which is where all of the transmitter's information is stored.

-sslpw <password>

specifies an SSL password that is used to initialize SSL when the transmitter starts. This option is the same as `-setsslpw` option, but it allows you to specify the password on the command line, instead of being prompted for it. (The `-setsslpw` option works with the `runchannel` program, but the `-sslpw` option works with tuner `-start`.)

Transmitter Administrator options

You can run Transmitter Administrator from the command line to perform many of its functions. The syntax for the command-line interface is as follows:

```
runchannel <Transmitter_Admin_CLI_URL>
  [<user_name>:]<password>@[<host>[:<port>]
  [<options>]
```

where

```
[<user_name>:]<password>@
```

authenticates access to the transmitter if required. If the transmitter you want to administer doesn't require a user name, you can specify only the password; if the transmitter uses anonymous administration, you can omit both the user name and password.

```
<host>[:<port>]
```

identifies the transmitter to administer. Be sure to specify the RPC port and not the port that tuners use (which is typically 5282). If you omit the port number, 7717 is used.

```
<options>
```

specifies the Transmitter Administrator command-line option.

The Transmitter Administrator options are described in the rest of this section. The options are `-help`, which displays the available options, or any of a set of additional options that are divided into the following categories:

- “The admin options” on page 144
- “The display options” on page 145
- “The ldap options” on page 146
- “The main options” on page 146
- “The publish options” on page 148
- “The repeating options” on page 149
- “The trans options” on page 152
- “The users options” on page 153
- “The workspace options” on page 154

Each of these categories (with the exception of display options) corresponds to a different part of the Transmitter Administrator user interface.

Note: If you have the Transmitter Administrator interface running when you run commands from the command line, the options you use are reflected in the interface window. For example, if you turn off subscription, the interface should immediately display the new status.

The admin options

`admin <options>`

specifies options that correspond to the Administration page in the Security section of Transmitter Administrator.

-anonymous

means anyone can administer the transmitter.

-anyuser

means any user in the database can administer the transmitter.

-emergency <password>

sets the emergency (back door) password for the transmitter.

-group <group>

is the name of the group of users who can administer the transmitter.

-inheritTunerPassword

sets the transmitter administration password to the password that's used to administer the tuner.

-passwd <password>

sets the transmitter administration password to the specified password.

-user <user_name>

is the name of the user who can administer the transmitter. This user name must exist in the LDAP directory, local database, or custom database.

The display options

display <options>

specifies options that display information related to transmitter administration.

-admin

displays the administration access permissions.

-allProperties

displays all transmitter properties and their values.

-database

displays the Users/Groups configuration.

-getProperty <property_name>

displays the specified transmitter property and its value.

-main

displays the following information about the transmitter:

Transmitter status

Port number

Workspace directory

Maximum concurrency

Upper and lower limits for files cache and index cache sizes

Compression status

-publish

displays the publishing status and publish hosts.

-trans

displays the subscription status and the input, output, and processing time-outs.

The ldap options

`ldap <options>`

specifies options that correspond to the Users/Groups page in the Security section of Transmitter Administrator.

`-adminRDN <name>`

sets the distinguished name for the administrator who can query the LDAP directory for users.

`-domain <name>`

sets the distinguished name for the entry in the LDAP directory where users and groups are stored.

`-passwd <password>`

specifies the password for the administrator who can query the LDAP directory.

`-schema marimba | netscape`

defines how the transmitter looks up users and groups in the LDAP directory.

marimba uses an organizational unit to define groups and user IDs to define the members of those groups. netscape uses an object class named `groupOfUniqueNames` or `groupOfNames` to define groups and `uniqueMember` or `member` to define members of those groups.

`-server <host>`

specifies the host for the LDAP directory server.

The main options

`main <options>`

specifies options that correspond to pages in the General section and the Performance section of Transmitter Administrator.

`-bind <host>`

is the host for a network interface.

`-changeRoot <directory>`

specifies the transmitter's workspace directory, in which the transmitter stores information and channels.

-compressionEnabled true | false
specifies whether the transmitter should compress all the files it sends.

-compressionLevel low | medium | high
specifies the level of compression the transmitter should use for the files it sends:

low means the compression is fast but the file size isn't reduced as much as on high (however the byte-savings difference is minimal). medium balances time and size. high means the file is compressed as much as possible, but for large files the process can take a long time and can use many CPU resources.

-concurrency <number>
specifies the number of clients (tuners) allowed to connect to the transmitter at one time.

-copyChannels <URL_of_channel_to_copy> [destination_path]
[segment][authentication]
copies the specified channel or directory (with contents) to the transmitter to which you are connected. <URL_of_channel_to_copy> is the source of the channel or directory. [destination_path] is the local path. If you do not specify a path, the channel or directory is copied to the root. [segment] limits copying to the specified segment.
[authentication] lets you access the source if access is restricted.

-diffsEnabled true | false
specifies whether the transmitter should use byte-level differencing, which allows the transmitter to send faster channel updates and to use less bandwidth. Instead of transferring entire files when updating channels, the transmitter uses byte-level differencing to send only the changed bytes.

-diffsMemory <kbytes>
specifies the amount of memory (in kilobytes) that the byte-level differencing algorithm uses.

-localClientsOnly true | false

limits the transmitter to accept only requests coming from tuners running on the same computer as the transmitter. Setting this option to true is useful when you are trying to troubleshoot problems or test a new set of channels before rolling them out to your company or department.

-minDiskFreePercent <percent>

specifies the percentage of disk space that the transmitter should keep free. To keep the specified amount of disk space free, the transmitter automatically deletes optional files, such as files in the index cache and files cache that only improve transmitter performance.

-on | -off

turns the entire transmitter on or off.

-port <port>

is the number of the port that the transmitter listens to. This is the port that tuners use when getting channels. Symphony Marimba Client Automation recommends using port 5282 or 80. Don't confuse this with the RPC port, which is 7717 by default.

-title <name>

changes the name of the transmitter as displayed in Transmitter Administrator.

The publish options

publish <options>

specifies options that correspond to the Publish section of Transmitter Administrator.

-addHost <host>

adds a computer to the list of computers (hosts) that can publish to the transmitter.

-delHost <host>

removes a computer from the list of trusted hosts.

`-mailto <address>`

is the email address to which the transmitter should send email when a channel is published.

`-noHosts`

clears any trusted host restrictions so that any host can publish to the transmitter.

`-on | -off`

turns publishing on or off.

`-passwd <password>`

specifies a password if required.

`-smtp <host>`

specifies the host for the email server.

The repeating options

`repeating <options>`

specifies options that correspond to the Replication section of Transmitter Administrator.

`-addMaster [<user_name>:]<password>@]<transmitter_URL>`

adds the transmitter specified by *transmitter_URL* to the list of master transmitters replicated by the slave transmitter. If the specified master transmitter requires authorization to send channels to a slave transmitter, specify *user_name* and *password*.

`-delMaster <transmitter_URL>`

removes the transmitter specified by *transmitter_URL* from the list of master transmitters replicated by the slave transmitter.

`-forcereplicate <#retries> <filter> [verbose]`

stops the replication module which in turn stops any ongoing replication process. This option specifies the channel that has to be replicated prior to replicating other channels. You can also specify the channel filter so that the option processes any URL that has a substring which matches the filter. After the specified channel is replicated, this command restarts the replication module and triggers the replication process.

<#retries> specifies the number of retries the command tries. You can specify 0 or any integer. You must specify this argument.

<filter> specifies the URL of the channel that the repeater or mirror should replicate from its master. You must specify this argument.

[verbose] produces more detailed status information, which is displayed in the console. This argument is optional.

-interval <minutes>

is the interval in minutes between updates to the replicating (repeater or mirror) transmitter from the master transmitter. This is the longest amount of time a replicating transmitter can have an outdated channel. Once a channel is updated on the master transmitter, the master transmitter does not forward requests to a replicating transmitter until the replicating transmitter downloads the updated channel.

-masterInRR true | false

controls whether the master transmitter is included in the round robin list when using that redirection strategy.

-masterIsBackup true | false

specifies that the master transmitter should handle requests from tuners if no repeaters can handle their requests.

-on | -off

turns replication on or off.

-redirClients true | false

controls whether the transmitter redirects clients, such as tuners, to active repeaters.

-refresh [<number_of_retries>] [verbose] [<filter_URL>]

instructs the Transmitter Administrator to synchronize the specified repeater or mirror transmitter with its master transmitter.

`<number_of_retries>` specifies the number of times the repeater or mirror transmitter should retry synchronizing with its master transmitter in case of intermittent network failures. `verbose` produces more detailed status information, which is displayed in the console. An exit status of 0 indicates operation success, while 1 indicates operation failure. `<filter_URL>` specifies the URL of the channel or folder that the repeater or mirror should replicate from its master. If you specify a folder, all the channels within that folder are replicated. If you do not specify this option, the repeater or mirror replicates all the content that it has permission to replicate from its master.

By default, the timeout is 15 minutes when making the initial contact to the master transmitter. If you want to change the timeout, you can use the `tuner` property

`marimba.transmitter.repeater.requested.timeout`. If you set it to 0, then no timeout is specified; otherwise, you specify the timeout in milliseconds.

Use the `-refresh` command-line option to wait until the transmitter you are refreshing is completely updated before the command returns. This option is useful when you are using the Deployment Manager (Server Management) to chain jobs.

Use Content Replicator to publish content to a master transmitter. The command returns when the publish is done.

Use the Transmitter Administrator `repeating -refresh` option to synchronize a mirror transmitter with its master. The command waits until the mirror transmitter is completely updated from the master before returning.

Use Content Replicator to install content from the mirror transmitter.

```
-strategy roundrobin | geographic | sbrp {-import <filepath> [-replication [<user>:]<password>] [-publish [<user>:]<password>] | -export <filepath> [-replication [<user>:]<password>]}
```

specifies the redirection strategy when using repeaters.

`roundrobin` means the transmitter starts with one repeater and continues delegating until it has delegated to all available repeaters. It then starts over with the first repeater. This is a useful strategy for a company with one or more work sites that are located close together.

`geographic` means the transmitter chooses a repeater that has the same time zone as the requesting tuner. This is a useful strategy for a company with many work sites around the world or in more than one time zone.

`sbrp -import <filepath> [-replication [<user>:]<password>] [-publish [<user>:]<password>]` imports the subnet-based repeater policy (SBRP) configuration file (`config.xml`) and publishes it to the master transmitter. `-import <filepath>` specifies the path and file name of the configuration file. `[-replication [<user>:]<password>]` (optional) specifies the replication user name and password. The user name can be omitted. `[-publish [<user>:]<password>]` (optional) specifies the publish user name and password. The user name can be omitted. For more information on the subnet-based repeater policy, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*.

`sbrp -export <filepath> [-replication [<user>:]<password>]` exports the subnet-based repeater policy (SBRP) configuration file (`config.xml`). `<filepath>` specifies the path and file name in which to export the configuration file. The export file path must be in a location that the Transmitter Administration command-line interface can access. Network paths are not supported. `[-replication [<user>:]<password>]` (optional) specifies the replication user name and password. The user name can be omitted.

`-timeout <minutes>`

is the number of minutes that tuners continue to direct requests to the repeater instead of the master transmitter.

The trans options

`trans <options>`

specifies options that correspond to the Performance section of Transmitter Administrator.

`-inputTimeout <seconds>`

is the number of seconds the transmitter waits before ending an idle request connection.

`-on | -off`

controls the transmitter service, which handles tuner requests for channels, the channel list page, and so on.

-outputTimeout <seconds>

is the number of seconds the transmitter waits before ending an idle reply connection.

-processingTimeout <seconds>

is the number of seconds the transmitter waits for a plug-in to respond to a request.

-thruput <Kbps>

limits the transmitter output to an exact number of kilobits per second. Specify the number of kilobits per second that the transmitter can use as throughput.

The users options

users <options>

specifies options that correspond to the dialog box presented when you click Edit on the Users/Groups page in the Security section of Transmitter Administrator. These options work only if the database isn't write-protected.

-add <user_name> <password>

adds a new user name and password to the database.

-addtogroup <user_name> <group>

adds a (previously created) user to a group in the database.

-database ldap | local | custom

specifies the database type.

-delete <user_name>

removes the user from the database.

-deletegroup <group>

removes the group from the database (although the users in the group remain in the database).

The workspace options

workspace <options>

specifies options that manipulate objects in the transmitter workspace.
The options are divided into three groups.

General options

These options work on all directories or channels in the transmitter workspace.

-delete <path_in_transmitter_workspace>

deletes the directory or channel.

-mkdir <path_in_transmitter_workspace>

creates a directory on the transmitter.

-setPerm <path_in_transmitter_workspace> <permission_type>
<authentication_type>

sets the permission and authentication values for a directory or channel.

- For <permission_type>, select subscribe, publish, or replicate.
- For <authentication_type>, select anonymous, anyuser, nobody, inherit, password <pass>, user <user>, or group <name>.

Node properties options

These options access the node properties in the transmitter workspace. These options are rarely used in day-to-day operations.

WARNING: Changing node properties can significantly affect transmitter operation.

-getProp <path_in_transmitter_workspace> <key>

returns the value for the specified key.

-setProp <path_in_transmitter_workspace> <key> <value>

sets the value for the specified key.

Channel properties options

These options access the channel properties and parameters inside the channel.

WARNING: Changing channel properties can significantly affect the channel.

-getParameters <channel_path> <output_file>

writes the values of the channel parameters to the specified file.

-getProperties <channel_path> <output_file>

writes the values of the channel properties to the specified file.

-setProperties <channel_path> <path_to_local_properties.txt_file>
[<path_to_local_parameters.txt_file>]

writes the specified properties file or both the specified properties and parameters files to the channel.

Tuner Administrator options

You can run Tuner Administrator from the command line to perform operations on a specified remote tuner (or on a channel on that tuner). The syntax for the command-line interface is as follows:

runchannel <Tuner_Admin_CLI_URL>

<operation> [<options>]

-tuner <host>:<port> -username <user_name> -password <password>

[-channel <channel_URL> [-channelUsername <user_name>]

-channelPassword <password>]]

where

-tuner <host>:<port> -username <user_name> -password <password>

specifies the target tuner and the user name and password for that tuner.

If the target tuner is configured to use SSL, you must use https:// before the host name (for example, -tuner https://client1:7717).

-channel <channel_URL> [-channelUsername <user_name>]

-channelPassword <password>]

specifies the channel on which to perform an operation. If this channel is on a transmitter that requires a user name and password, specify them with the `-channelUsername` and `-channelPassword` options.

`<operation>`

specifies the operation Tuner Administrator is to perform.

The Tuner Administrator operations are described in the rest of this section. Some of these operations get additional information from command-line options. With a few exceptions, each operation requires that you specify a channel in the command (with `-channel` and possibly `-channelUsername` and `-channelPassword`).

`-denyCert`

denies the channel's certificate.

`-disableSSL`

disables SSL on the tuner.

`-enableSSL <SSLCertificateName> <SSLpwd> <saveSSLPwd>`
`<sslClientAuth>`

enables SSL on the tuner where

`<SSLCertificateName>` specifies the certificate name

`<SSLpwd>` specifies the certificate password

`<saveSSLPwd>` specifies whether or not to save the password. The value can be true or false. The default value is false.

`<sslClientAuth>` specifies the client authentication type. The value can be none, require, or request. The default value is none.

`-get`

displays the value of the tuner property or channel property specified in the `-property` option or, if no `-property` option is included, lists all the channels on the tuner. If you use this operation to get a channel property, you must specify a channel in the command.

`-getCert`

displays the channel's certificate.

`-getLicense`

displays the channel's license text.

`-getLogs <options>`

generates all the log files of a target tuner or only the log files of a specific channel in a zip file.

-output <output_file>

The location of the zip file where the logs can be retrieved. For example, C:\TunerLogs.zip.

If you want to download tuner related logs, use the following syntax:

```
runchannel  
<URL_of_the_Tuner_Administrator_command_line_channel> -  
getlogs -tuner <remote_tuner_IP:port> -username <user_name> -  
password <password> -output <output_file>
```

If you want to download the logs related to a specific channel, use the following syntax:

```
runchannel  
<URL_of_the_Tuner_Administrator_command_line_channel> -  
getlogs -tuner <remote_tuner_IP:port> -username <user_name> -  
password <password> -channel <URL_of_the_channel> -output  
<output_file>
```

-getSchedule

displays the channel's update schedule.

-help

lists and briefly describes all the command-line operations and options available for Tuner Administrator.

-remove

deletes the channel from the tuner.

-restart

restarts the tuner. You don't need to specify a channel in the command.

-savesslpw

prompts for and saves the tuner's SSL password. You don't need to specify a channel in the command.

-set

sets the tuner property or channel property specified in the `-property` option to the value specified in the `-value` option. If you use this operation to set a channel property, you must specify a channel in the command.

-setCert

sets the SSL certificate for the tuner to the certificate with a unique ID specified in the `-value` option.

-setSchedule

sets the channel's update schedule to the schedule specified in the `-value` option.

-start

-stop

starts or stops the channel.

-subscribe

subscribes the tuner to the channel.

-unsubscribe

unsubscribes the channel from the tuner.

-update

updates the specified channel.

-updateFrom

changes the URL from which the channel gets its updates to the URL specified in the `-newUrl` option.

The additional command-line options are described in the rest of this section.

-arguments “<argument>” | ”<argument1> <argument2>...”

specifies arguments to pass to the channel when it starts. If you specify one argument, the argument should be enclosed within quotation marks (“”). If you specify more than one argument, the arguments should be separated by a space and the entire list of arguments should be enclosed within quotation marks. For example:

-arguments "-nogui -q -f C:\Temp\settings.txt"

None of the arguments can contain spaces (since the arguments are delimited by a space). So, a file name specified in an argument cannot contain a space within it (for example, C:\My Temp\settings.txt).

`-certPassword <password>`

is the client-certificate password for the access control transmitter.

`-newUrl <channel_url>`

is the new URL from which the channel gets its updates for the

`-updateFrom` operation.

`-property <property_name>`

is the tuner property or channel property on which a `-get` or `-set` operation is to perform.

`-trust <subject> <issuer>`

means that any channel using a certificate that matches

`<subject>|<issuer>` is trusted by the tuner. Either `<subject>` or `<issuer>` can be the string "any", which allows the tuner to trust any subject or issuer. For example:

- "cn=Acme, Inc."|"ou=VeriSign Trust Network" refers to a specific certificate.
- "cn=Acme, Inc."|"any" refers to any Acme certificate.
- "any"|"ou=VeriSign Trust Network" refers to any certificate issued by VeriSign.
- "any"|"any" means any channel with a certificate is trusted.

`-value <new_value>`

is the new value for the following operations: `-set` (a tuner or channel property value); `-setCert` (the certificate's unique ID), or `-setSchedule` (a channel schedule). A channel schedule's value can be frequently, never, hourly, daily, or weekly. (The schedule for a tuner is set through the tuner property `marimba.schedule.filter`.)

For example:

```
runchannel http://mytrans/TunerAdministrator -start
-tuner server:7717 -username admin -password secret
-channel http://transmitter:5282/test
```

starts the test channel on the tuner on the host server.

```
runchannel http://mytrans/TunerAdministrator
-tuner server -username admin -password secret
-set -property marimba.tuner.admin -value bob,plain:a6t5p8
```

sets the tuner's marimba.tuner.admin property to bob,plain:a6t5p8.

```
runchannel http://mytrans/TunerAdministrator
-tuner server -username admin -password secret
-get
```

lists the channels on the tuner.

Using Tuner_ns commands from the command-line

You can use the tuner_ns.exe executable to run commands on a running tuner in Windows 7 and Windows Vista when you are logged in as non-admin user. This can be used for running commands such as subscribe, unsubscribe, start, and stop channels. You cannot use tuner_ns.exe executable to start a tuner. You can use the tuner_ns.exe instead of the runchannel.exe.

This applies only to the tuner running in a newly created user account with administrator privileges and not the built-in administrator account. When you use tuner.exe to start a package or a channel from the command-line, the User Access Control (UAC) is displayed. However, the UAC is not displayed if you use tuner_ns.exe.

The following table shows the commands which you can use as an Administrator and as a non-Administrator:

Runchannel	Administrator	Non-Administrator	tuner_ns	Administrator	Non-Administrator
subscribe	Yes	Yes	subscribe	Yes	Yes
nosubscribe	Yes	Yes	nosubscribe	Not Applicable	Not Applicable
update	Yes	Yes	update	Yes	Yes
noupdate	Not Applicable	Not Applicable	noupdate	Yes	Yes
rpc	Yes	No	rpc	Yes	No
norpc	Yes	No	norpc	Yes	No
help	Yes	Yes	help	Not Applicable	Not Applicable

Runchannel	Administrator	Non-Administrator	tuner_ns	Administrator	Non-Administrator
p	Yes	Yes	primary	Yes	No
quit	Yes	Yes	quit	Yes	Yes
timeout	Yes	Yes	timeout	Not Applicable	Not Applicable
remove	Not Applicable	Not Applicable	remove	Yes	No
ws	Yes	No	ws	Yes	Yes
start	Not Applicable	Not Applicable	start	Yes	Yes
stop	Not Applicable	Not Applicable	stop	Yes	Yes
Command-line for packages					
-subscribe -start	Yes	Yes	-subscribe -start	No	No
install	Yes	Yes	install	No	No
remove	Yes	Yes	remove	Yes	Yes
verify	Yes	Yes	verify	No	No
version	Yes	Yes	version	No	No
repair	Yes	Yes	repair	No	No
silent	Yes	Yes	silent	No	No

Generating the heap dump of the tuner

-getheapdump

generates the heapdump of the Tuner. On Unix machines the heapdump is generated in the bin folder of the Tuner workspace directory. On Windows machines the heapdump is generated in the Tuner folder of the Tuner workspace directory.

For example:

On Windows:

tuner.exe -getheapdump

On Linux:

./tuner -getheapdump

Chapter

2

Tuner properties

You can use tuner properties to qualify the appearance, dialup behavior, log support, generic http and proxy behavior, event scheduling, security details, and runtime properties of a tuner.

The following topics are provided:

- Overview of tuner properties (page 164)
- List of tuner properties (page 164)
- Tuner Enhancements (page 225)

Overview of tuner properties

Many tuner property descriptions in this section refer to aspects of the tuner that are discussed in other documentation. In particular, the tuner command-line options are described in “Command-line options” on page 19. The tuner properties that you can set when creating a tuner installer during setup and deployment are described in this section.

You can set some of these tuner properties after the tuner is installed. The properties are stored in the `prefs.txt` file in the tuner’s workspace. For more information about the tuner’s workspace, see the *Symphony Marimba Client Automation CMS and Tuner User Guide*, available on the Marimba Channel Store. You can set the tuner properties by using Tuner Administrator or by directly editing the `prefs.txt` file. Restart the tuner for any changed tuner properties to take effect.

A read-only `properties.txt` file is located in the tuner’s workspace (usually, `C:\Program Files\Marimba\Tuner\lib\tuner\properties.txt` on Windows). The tuner properties listed in this file are the defaults, but when the tuner runs, the tuner properties in the `prefs.txt` file override these defaults.

List of tuner properties

This section lists the tuner properties.

Note: The following list does not include all the possible tuner properties.

You can change some tuner properties through the graphical user interface (GUI), while Marimba modules set others, and changing them manually is not recommended.

`channel.defer.idle`

specifies that when you set the value of this channel property to `true`, when this channel is started, the tuner resets the idle timer of the Windows operating system, and prevents the computer from sleeping. The tuner un-blocks sleep mode of the computer when this channel stops running.

Note: This property does not prevent you from manually forcing a sleep or hibernate mode through the power button or through the Startup menu on Windows.

marimba.auditlog.enabled

enables and disables audit logging for the tuner and channels in the tuner. By default, this property is set to true. You can disable audit logging by setting this property to false. Any channel events that occur while audit logging is disabled are not recorded in the log files. However, any previous log entries in the audit log files are not deleted.

Valid value: true or false

Default value: true

marimba.bandwidth.max

is the maximum amount of bandwidth that the tuner can use for all channel requests including manual updates and requests initiated by the scheduler or Policy Management. (However, Content Replicator ignores this property during install operations and uses its own -rate option.) If this property is not set, then the tuner does not limit the bandwidth used.

You can specify the property as an absolute value in kilobits per second or as a percentage of a specific bandwidth (28800, 56000, 128000, 1544000, 10000000, 44736000, 100000000, 155000000) in bits per second.

For example:

20 (20 kilobits per second)

10/28800 (10% of 28800 bits per second).

See `marimba.schedule.polite.speed`.

Valid value: an integer or percentage

Default value: null

marimba.browser.home

is the URL of the home page of the browser specified by `marimba.browser.url`.

Valid value: a URL

Default value: `http://www.marimba.com/browse`

`marimba.browser.url`

is the URL of the channel that can act as a browser inside the tuner.

Valid value: a URL

Default value: `http://trans.marimba.com/Browser`

`marimba.channelmanager.startmenu`

adds a shortcut entry to the Windows Start menu for each channel to which the tuner is subscribed when set to true. When you click the Start menu, you see a Channel option that contains a list of channels. Under Channels, the channels are sorted by category and name. Setting this property to `false` removes the channel shortcuts from the Start menu. This property is used for Windows platforms only.

When you subscribe to a channel, the channel's name is added as a shortcut to the list of channel on the Start menu. When you delete a channel, the entry is removed. If you unsubscribe from a channel, the shortcut remains available, and when you select the unsubscribed channel, it gets subscribed and updated before it runs.

Valid value: `true` or `false`

Default value: `false`

`marimba.channelstore.shortcut.usetunerns`

specifies whether `tuner_ns.exe` should be used for launching Channel Store through desktop shortcut. By default, the desktop shortcut triggers Channel Store through `tuner.exe`, which runs in admin privileges. In cases where the user lacks administrator privileges, setting this property to a value of `true` will cause the shortcut to be launched with invoker privileges.

Note: To implement this, you must set this property before the Channel Store is deployed on a tuner. If the Channel Store channel is already subscribed on a tuner, the channel needs to be removed or unsubscribed from the tuner and then re-subscribed to take the value specified for this property.

Valid value: `true` or `false`

Default value: `false`

`marimba.console.codepage`

specifies the code page to use for the ANSI character set. For example, if French characters are not appearing properly on the command line, you can set the code page to 1252 and set the DOS font to Lucida Console.

Valid value: a code page number

Default value: none

The `marimba.dialup.<option>` properties apply to the Windows dial-up networking feature and to Windows platforms only.

`marimba.dialup.enabled`

indicates whether the user is dialing up to connect to the network. If this property is not set to true, then the rest of the `marimba.dialup.<option>` properties are ignored.

Valid value: true or false

Default value: false

`marimba.dialup.password.<server>`

is the password specified in the named dial-up connection icon that is answered by a specified server `<server>`. You usually set the password by entering a string from the Channel Manager or Tuner Administrator.

Valid value: a string entered from the Channel Manager or Tuner Administrator, or a Base64-encoded string when directly editing the `prefs.txt` file

Default value: null

`marimba.dialup.server`

is the currently selected dial-up server. If you have more than one dial-up connection specified in a comma-separated list, this property displays the currently used one.

Valid value: a string

Default value: null

`marimba.dialup.servers`

is a comma-separated list of the available dial-up servers.

Valid value: a string

Default value: none

`marimba.dialup.timeout`

is the timeout for the dial-up connection (in minutes). The Windows operating system is responsible for enforcing this value.

Valid value: an integer

Default value: depends on the Windows operating system

`marimba.dialup.user.<server>`

is the user name associated with the dial-up server `<server>`.

Valid value: a string

Default value: null

`marimba.http.agent`

is the identifying string set in the User-Agent field when the tuner makes HTTP requests of a server.

Valid value: a string

Default value: Castanet Tuner 30a2

`marimba.http.port`

is the port where the tuner can act as a proxy for an HTML channel. This functionality is needed so that a local browser can use web pages stored in the tuner's channel. This is an entirely separate feature from the external proxy that the tuner can use when it makes HTTP connections.

Valid value: an integer

Default value: 5283

`marimba.http.showtip`

indicates whether the tuner reminds users to set the browser's proxy to point at the tuner. This reminder appears when the user starts an HTML channel only if `marimba.http.showtip` is true and `marimba.browser.url` is not set.

Valid value: true or false

Default value: true

`marimba.intro.url`

is the URL of the intro channel that runs only once, the first time the tuner is run.

Valid value: a URL

Default value: `http://trans.marimba.com/IntroducingCastanet`

`marimba.inventory.concat.tunerid`

appends the tuner ID to the mac ID on all platforms when set to `true`. This property is useful when using the Inventory module on partitioned machines, which can share the same mac ID (and can cause the inventory scan in the database to be overwritten).

Valid value: `true` or `false`

Default value: `false`

`marimba.inventory.customscanner.timeout`

represents the timeout period for the custom scanner (scanner extension) to complete its scan. If Scanner Service hangs, or seems to run indefinitely, it eventually automatically times out after the number of minutes specified by this property.

Valid value: an integer (time in minutes)

Default value: 60

`marimba.inventory.nativescanner.timeout`

represents the timeout period for the native scanner (`winscan.exe`) to complete its Windows-specific system/hardware scan. If Scanner Service hangs, or seems to run indefinitely, it eventually automatically times out after the number of minutes specified by this property. This property is used for Windows platforms only.

Valid value: an integer (time in minutes)

Default value: 10

`marimba.inventory.plugin.enablepatchhistory`

Enables the patch history related table data insertion when set to '`true`' at tuner level where the Inventory plug-in is hosted. This property is applicable from 7202 version.

Default value: `false`

Warning: If you set this property '`true`' and data is populated in database, performance or upgrade issues will occur during an upgrade to a later version of the database schema.

`marimba.inventory.plugin.dbinsert`

specifies whether the Inventory plug-in inserts scan reports into the database. You must stop the Inventory plug-in before you set this property, and restart the plug-in after you set this property.

Valid value: true (reports are inserted) or false (reports are not inserted)

Default value: true

`marimba.inventory.plugin.forwardURL`

specifies the URLs for the mirrors or master transmitter in a LAN (Local Area Network) to receive scan reports from mirrors in a WAN (Wide Area Network). You must stop the Inventory plug-in before you set this property, and restart the plug-in after you set this property. To use this property, first set the `marimba.inventory.plugin.dbinsert` property to false.

Valid value: A list of comma-separated URL addresses

Default value: null

`marimba.inventory.plugin.forward`

If this tuner property is set to "true", the Inventory plugin forwards the scan report to Inserter or Master Transmitter's Inventory Plugin.

If this tuner property is set to "false", the Inventory plugin tries to insert the scan report directly into the database.

Default value: true

Note: However, when "marimba.inventory.plugin.forward" property is set to false, it can be over-ridden by the tuner property "marimba.inventory.plugin.dbinsert". For more information on this tuner property, see `marimba.inventory.plugin.dbinsert`.

`marimba.inventory.scanner.prop.scanner.comp.softwaretitle.scan`

Note: Enables software title data insertion when set to 'true' in the Tuner prefs.txt file where plug-in is hosted. When set to false, this property disables the functionality. This property applies to version 7.5 and later.

Default value - true

`marimba.inventory.smbiosbiosversion`

When set to 'true,' sets the inventory Service captures the value for SMBIOSBIOSVersion. When set to 'false,' the Inventory Service captures the version attribute for win32_BIOS class in WMI.

Default value: false

marimba.launch.args

specifies command-line options for the tuner to use when it starts. Any input required for options should appear in quotation marks. This property is used for Windows platforms only.

For example:

```
marimba.launch.args=-nointro -nodisplay -noproxy -trust  
"cn=Marimba, Inc.;c=US" any
```

In this example, the `nodisplay` option prevents the primary channel from showing a grant/deny dialog box when the tuner starts.

Valid value: a string

Default value: null

marimba.launch.console

specifies whether to allocate a console window, or optionally to redirect output to a file. This property is used for Windows platforms only.

Valid value: yes, no, or a filename where output is redirected

Default value: null

marimba.launch.console.hide (deprecated)

specifies whether to allocate a console window, and then hide it right away. This property is intended as a workaround for a JDK 1.3.1 limitation. On Windows NT using JDK 1.3.1, when the tuner runs as a service and a user logs off, a Wait/End task dialog is shown. This problem does not occur if a console window is allocated. When this property is set to yes, a console is allocated and then hidden immediately. The effect is that when a tuner starts or comes out of minimal mode, a window flashes. The window does not flash if the tuner runs as a service without desktop interaction. This property takes effect only when the tuner is running as a service and using JDK 1.3.1.

Note: This property has been deprecated. Use the `-Xrs` Java launch argument instead.

Valid value: yes or no

Default value: null

`marimba.launch.defaultWakeupTime`

specifies the default wake-up time to use when the tuner minimized, but it did not leave a wake-up schedule (indicating no channel events are pending).

Valid value: an integer (time in minutes)

Default value: 1440 (24 hours)

`marimba.launch.inheritPath`

specifies whether the system variable PATH (and the directory paths specified in that variable) is included in the paths that the Java Virtual Machine (JVM) can see. For example, if you have a custom channel and you want to use functions in .dll and .class files with directory paths that are specified in PATH, then set this property to yes.

Valid value: yes or no

Default value: no

`marimba.launch.javaArgs`

specifies a list of arguments to pass to the Java VM. You should separate arguments by spaces. You can use this property to change settings such as the heap size of the VM. Unlike most of the `marimba.launch.*` properties, which work only on the Windows platform, this property works on the UNIX platform.

Valid value: a string

Default value: null

`marimba.launch.logFile`

specifies whether to create a `launch.log` file in the tuner's workspace. The `launch.log` file contains events from the minimized tuner.

There is no default value to this property. By default, the `launch.log` file is always created in the tuner workspace only on the Windows platform.

When you set the value of the `marimba.launch.logFile` property to "no", no `launch` file is created in the tuner workspace. Setting any value other than "no" to the `marimba.launch.logFile` property does not affect the default behavior of creating the `launch` file.

`marimba.launch.logLevel`

sets the verbosity level for Windows event log messages from 1 (lowest level) to 3 (highest level). By default, a debug file called `launch.log` is created regardless of the verbosity setting of the `marimba.launch.logFile` property. The file is created in the tuner's workspace directory for any value other than "no" that you specify for the `marimba.launch.logFile` property. The file is not created only if you specify "no" for the `marimba.launch.logFile` property. The `launch.log` file is created every time the tuner minimizes, so the file stays reasonably small.

Valid value: an integer from 1 to 3

Default value: 1

`marimba.launch.maxWakeUpTime`

determines the maximum time the tuner minimizes before starting the Java VM again. If the tuner minimizes and the wake-up time is greater than the value set for this property, then the wake-up time is changed to the maximum value set for this property. A value of 0 means the wake-up time remains unchanged.

Using this property can cause a channel to update more often than specified in its schedule. This can happen if the update schedule for the channel is greater than the wake-up time, and the channel is the next one to be updated. If maintaining the channel's update schedule is important, then do not use this property.

Valid value: an integer (time in minutes)

Default value: 0

`marimba.launch.multiprocessor`

controls whether the tuner runs in multi-processor mode on Windows machines that have multiple processors.

Valid value: yes or no

Default value: no

`marimba.launch.NTServiceArgs`

specifies a list of command-line arguments (separated by spaces) that are passed to the tuner. The arguments are used when the tuner is running as a service on Windows.

Valid value: a string

Default value: null

`marimba.launch.redirect`

specifies whether to redirect output to a file. The default is `false`. If the property is set to `true`, standard output and errors are redirected to a file inside the tuner's workspace file (`/logs/tuner.log`). This property does not affect tuners that run without user interaction (through `start.sh`); those tuners continue to redirect standard output and errors (`stdout/stderr`). This property works on the UNIX platform.

Valid value: `true` or `false`

`marimba.launcher.autorestart`

is the time (in seconds) after which the tuner attempts to minimize its memory footprint by shutting down the VM (also called going into minimal mode). This is done to free up as much memory as possible when the tuner is not in use. If the value is 0, this feature is disabled.

For historical reasons, a value of 0 disables this feature. However, such usage is discouraged, and the recommended approach is to set `marimba.tuner.minimize` to `false`. Future releases may not support disabling tuner minimization by setting this property.

Valid value: an integer (time in seconds)

Default value: 60

`marimba.ldap.admanagementdomain`

specifies the domain for a machine (for example, `marimba.ldap.admanagementdomain=acme.com`).

This property is useful when you want to use machines that are not in a domain, such as UNIX machines, for collections in Policy Manager or Report Center. You must set this tuner property on the following:

- machines where Policy Manager, Report Center, and the CMS are running
- endpoint machines where the Policy Service is running
- master transmitter and repeater machines running on Sun Solaris or Linux

The Global Catalog must also be enabled in the domain where the Collections container is located.

Valid value: a string representing a domain where the Collections container is located

`marimba.ldap.connectiontimeout`

specifies the timeout value when connecting to a directory service. Be cautious when increasing the value for this property because increases can require a large amount of memory and affect the overall performance of the system.

Valid value: an integer (time in seconds)

Default value: 60

`marimba.ldap.querytimeout`

specifies the timeout value when querying a directory service for data. Be cautious when increasing the value for this property because increases can require a large amount of memory and affect the overall performance of the system.

Valid value: an integer (time in seconds)

Default value: 60

`marimba.ldap.srvdnsserver`

specifies a list of DNS servers that can be used to look up SRV records. This property is useful when it is not possible to change the `resolve.conf` file for a machine. However, machines running the plug-in and Policy Manager must be able to resolve the host name of the returned domain controllers and global catalogs.

Valid value: a comma-separated list of DNS servers

`marimba.logs.enabled`

enables and disables logging for the channels in the tuner. By default, this property is set to `true`. You can disable logging by setting this property to `false`. Any channel events that occur while logging is disabled are not recorded in the log files. However, any previous log entries in the log files are not deleted.

Valid value: `true` or `false`

Default value: `true`

`marimba.logs.rangefilter`

allows you to filter out certain log messages from the tuner or channel logs. You can use this property to specify a list of log IDs or log ID ranges (separated by a comma) that are not logged. You can specify ranges using the following form:

<starting log ID>-<ending log ID>.

Note: If the tuner is started with the `-v` (verbose mode), then this property is not honored.

For example:

`marimba.logs.rangefilter=1748, 1000-1100, 1500-1510, 1747`

Valid value: a list of log IDs or log ID ranges, separated by a comma (see “Logging codes” on page 307).

Default value: none

`marimba.logs.roll.policy`

specifies the policy for rolling the log files for all the channels in the tuner, but not the tuner itself. (For information about the property for rolling tuner log files, see `marimba.tuner.logs.roll.policy`.) This property requires a tuner restart to take effect.

If you choose `manually` or `never`, the log entries are recorded in a single file, which is never automatically rolled. If you choose `bysize`, the log file is rolled automatically when it reaches the size specified in `marimba.logs.roll.size`.

Valid value: `hourly`, `daily`, `weekly`, `monthly`, `yearly`, `manually`, `never`, `bysize`

Default value: `bysize`

`marimba.logs.roll.size`

specifies the size in kilobytes the log file must reach before it is rolled automatically. The value of this property is used when `marimba.logs.roll.policy` is set to `bysize`.

Valid value: an integer (size in kilobytes)

Default value: 32

`marimba.logs.roll.versions`

specifies the number of previously rolled log files for the channel that can exist.

Valid value: an integer

Default value: 1

`marimba.network.detection.address`

specifies the IP address (not a host name) of the host that you want to ping. The address of a router is a good choice for the value to use (for example: `marimba.network.detection.address=216.200.61.168`). This property must be used with `marimba.network.detection.policy=ping`.

Valid value: a string representing an IP address

Default value: null

`marimba.tuner.nw.detectinterval`

specifies the time in milliseconds after which the tuner resumes detecting when a computer wakes up from the sleep or hibernate mode. To disable the

`marimba.tuner.nw.detectinterval` property, you can set the value of this property to -1. By default, this feature is disabled and the default value of this property is set to -1.

Valid value: milliseconds

Default value: -1

`marimba.network.detection.delay.offline`

specifies how frequently (in seconds) the tuner polls if a polling-based network detection policy (for example, ping or multicast) is in effect. This property applies when the network is not detected.

Valid value: an integer

Default value: 30

`marimba.network.detection.delay.online`

specifies how frequently (in seconds) the tuner polls if a polling-based network detection policy (for example, ping or multicast) is in effect. This property applies when the network is detected.

Valid value: an integer

Default value: 60

`marimba.network.detection.mcgroup`

specifies the multicast address. You can set this property in the `prefs.txt` file.

For example: `marimba.network.detection.mcgroup=224.0.0.1`

`marimba.network.detection.ping.retries`

specifies the number of times to try the ping before concluding there is no network available. This property must be used with `marimba.network.detection.policy=ping`.

Valid value: an integer

Default value: 3

`marimba.network.detection.policy`

determines the tuner's network detection behavior. The default value for this property is platform-specific. On UNIX platforms (such as Tru 64) the default is `on`. All other platforms default to `detect`, as does the behavior if the property is not set. Setting the property to `detect` tells the tuner to use multicast to detect the network. This property is available only in Tuner 4.5.0.3 or later releases.

In Tuner 4.6 or later releases, you can set this property to `ping` to use an alternative network detection policy. This policy works by attempting to *ping* a specified host. It is useful when trying to work around a limitation on Windows, where if the host is configured with a static IP address, the status can be mistakenly reported as “network detected.” If you use the `ping` network detection policy, you must also specify the following tuner properties:

- `marimba.network.detection.address`
- `marimba.network.detection.ping.retries`

Valid value: a string, such as `on`, `detect`, or `ping` (see the description above)

Default value: Platform specific (see the description above)

`marimba.primary.url`

is the URL of the tuner's primary channel, which is the first channel started when the tuner starts. See the tuner command-line options `-primary` and `-noprimary`.

Valid value: a URL

Default value: `http://trans.marimba.com/ChannelManager`

`marimba.proxy.bypass`

specifies whether the tuner tries to bypass the proxy if a connection error occurs. This behavior applies while updating a channel, subscribing to a channel, and when channels make any type of URL connections (for example, connecting to a transmitter). See `marimba.proxy.url.bypassalways`.

Valid value: `true` or `false`

Default value: `false`

`marimba.proxy.enable`

enables proxies. This property is typically used as a temporary switch for setting up and testing proxies.

Valid value: `true` or `false`

Default value: `false`

`marimba.proxy.exceptions`

is a comma-delimited list of host suffixes for which the proxy settings do not apply. For example, setting this property to `havefun.com,marimba.com` means that any HTTP requests to `havefun` or `marimba` do not go through a proxy.

Valid value: a comma-delimited list of host suffixes

Default value: `null`

`marimba.proxy.http.host`

`marimba.proxy.https.host`

specifies the HTTP-proxy host or HTTPS-proxy host (that is, a proxy to which you are making an SSL connection) using the following form:

`<host>:<port>`

See `marimba.proxy.exceptions` and `marimba.proxy.notforunqualifiedhosts`.

Valid value: a string in the form `<host>:<port>`

Default value: `none`

`marimba.proxy.http.list`
`marimba.proxy.https.list`

specifies a list of HTTP-proxy hosts or a list of HTTPS-proxy hosts (that is, proxies to which you are making an SSL connection) for proxy failover using the following form:

`<host1>:<port1>;<host2>:<port2>;<host3>:<port3>`

where `<host1>` is the host name of the first proxy, `<port1>` is the port number for the first proxy, and so on.

If proxy authentication is required, use `marimba.proxy.http.password` or `marimba.proxy.https.password` to specify the proxy password. The proxy password must be the same for all the proxies in the list. See `marimba.proxy.exceptions` and `marimba.proxy.notforunqualifiedhosts`.

Valid value: a string in the form

`<host1>:<port1>;<host2>:<port2>;<host3>:<port3>`

Default value: none

`marimba.proxy.http.password`
`marimba.proxy.https.password`

is the password that is used when connecting to an HTTP proxy or HTTPS proxy. Use this property only if proxy authentication is required. The valid value for this property is a Base64-encoded string of the following form:

`<user_name>:<password>`

where `<user_name>` is the user name, and `<password>` is the proxy password. Type the user name and password as shown, encrypt it to a Base64 encoded string, and then copy and paste that string into the property setting.

Note: The proxy password must be the same for all the proxies in the list.

Valid value: a Base64-encoded string of the form

`<user_name>:<password>`

Default value: none

```
marimba.proxy.http.table
marimba.proxy.https.table
```

specifies the HTTP proxy or HTTPS proxy (that is, a proxy to which you are making an SSL connection) that the tuner connects to instead of connecting directly to the specified transmitter. You can use this property when you have groups of transmitters that belong to different subnets, and you want some subnets to be associated with one proxy and the other subnets to be associated with another proxy. You specify the netmask, the IP address (that specifies the numbers you want to match with the actual IP addresses of the transmitters), and the IP address and port number of the proxy. This property has the following form:

`<netmask>,<ip_addr>,<proxy>`

The value of `<netmask>` is used to find the significant bits in `<ip_addr>` that are matched to the IP address of the transmitter. If the transmitter's IP address matches the value `<ip_addr>`, then the proxy specified by the value `<proxy>` is used. For `<proxy>`, you can specify one of the following values:

- IP address and port number with the form `<ip_addr>:<port>`
- 0.0.0.0—the tuner does not connect to a proxy. It connects directly to the transmitter.

For example, you can set up your table as:

```
marimba.proxy.http.table=255.255.0.0,172.16.0.0,172.16.1.4:600
0
```

In this example, the netmask specifies that the first two numbers of the IP address are matched with the IP address of transmitters. So, if the tuner has to connect to the transmitter with the IP address 172.16.4.4, the tuner is directed to the specified proxy.

Valid value: a list of IP addresses (see description above)

Default value: null

```
marimba.proxy.notforunqualifiedhosts
```

indicates whether the proxy connects to hosts with names that are not qualified with domain names. For example, if this property is true, you are able to connect to `www.marimba.com` through the proxy, but not to `www.`

Valid value: true or false

Default value: false

`marimba.proxy.socks.host`

specifies the SOCKS-proxy host using the following form:

`<host>:<port>`

If SOCKS is turned on, all socket connections use it, so other proxy settings can still apply. There is no granularity in selecting which protocols use the SOCKS setting. If this property is set, all the tuner connections use SOCKS.

Valid value: a string in the form `<host>:<port>`

Default value: null

`marimba.proxy.trust`

indicates whether security is relaxed by allowing the proxy to resolve hosts. This is necessary if your firewall doesn't allow DNS lookups of hosts outside the firewall.

Setting this property to true makes it impossible to use `marimba.security.trusted.transmitters` with IP addresses, although it's still possible to use DNS host names.

Valid value: true or false

Default value: false

`marimba.proxy.url.bypassalways`

specifies whether the tuner always bypasses the proxy when channels make any URL connections (for example, connecting to a transmitter). Unlike `marimba.proxy.bypass`, this property does not affect updating or subscribing to channels.

Valid value: true or false

Default value: false

`marimba.rc.fdcc.enable`

displays the USGCB Reporting menu item in the Application menu of CMS. To display the USGCB Reporting menu item, Set the value of this property to true in CMS tuner's prefs.txt file. By default, from BBCA 8.2.02.001, the USGCB Reporting menu item does not appear in the Application menu list.

Valid value: true or false

Default value: false

`marimba.reboot.interact`

specifies whether an alert message warning users about a reboot is displayed.

Valid value: true or false

Default value: true (an alert message is displayed)

`marimba.reboot.interact.allowSnooze`

specifies whether users are allowed to postpone a reboot. If set to true, a dialog box is displayed allowing users to postpone a reboot. If set to false, an alert dialog box is displayed that shows a timer counting down to the reboot.

Valid value: true or false

Default value: true

`marimba.reboot.interact.snooze.maxTime`

specifies the maximum amount of time (in milliseconds) that you want to allow users to postpone a reboot.

Note: If the value is set to the default, there is no timeout and the reboot does not occur.

Valid value: an integer (time in milliseconds)

Default value: 0 (no maximum amount of time).

`marimba.reboot.interact.timer`

specifies the amount of time in seconds to wait or count down before rebooting. This value is the starting time displayed in the alert dialog box if you set `marimba.reboot.interact.allowSnooze` to false. If you set this value to 0, the machine reboots immediately. However, the `marimba.reboot.interact.timer` property works only if `marimba.reboot.timeout` property does not exist.

Valid value: an integer (time in seconds)

Default value: 60

`marimba.reboot.never`

specifies whether to automatically reboot the machine if a reboot is required (for instance, after installing patches). Set the value to true if you do not want to automatically reboot the machine.

Valid value: true or false

`marimba.reboot.timeout`

specifies the amount of time in seconds to wait or count down before rebooting. This property when set has precedence over the `marimba.reboot.interact.timer` property.

`marimba.schedule.dialup`

indicates whether the scheduler updates channels even if it must turn the modem on first. If set to `false`, the scheduler does not run when the network is inactive. If set to `true`, the scheduler looks at `runtime.network.online`; otherwise, it looks at `runtime.network.enabled`.

Valid value: true or false

Default value: `false`

`marimba.schedule.filter`

indicates the periods of time during which the scheduler is active.

Examples of the form this property's value can take follow:

never

anytime on mon+tue

at 4:00AM on mon+tue

between 09:00AM and 05:00PM on mon+tue+wed

For more information, see “Syntax for the schedule string” on page 259.

Valid value: a string

Default value: `anytime on sun+mon+tue+wed+thu+fri+sat`

Blackout periods and the `marimba.schedule.filter` property. When you specify a blackout period using Policy Manager, Policy Service appends the time period you specify to the tuner property `marimba.schedule.filter` in the endpoint tuners' `prefs.txt` file. For example, if an endpoint tuner's original update schedule for channel is `anytime` on any day of the week, and then you use Policy Manager to set a blackout period from 9 AM to 5 PM, the tuner property is set to the following value:

`marimba.schedule.filter=ANYTIME on mon+tue+wed+thu+fri+sat+sun
BLACKOUT 9:00AM-5:00PM`

If you later select the No blackout period option in Policy Manager, the original channel update schedule set at the endpoint tuner is used. In the example given above, only BLACKOUT 9:00AM-5:00PM is removed from the value for the marimba.schedule.filter. The tuner property that affects the blackout period (marimba.schedule.filter) is not deleted from the endpoint. You can delete that tuner property explicitly using the Tuner and Package Properties page in Policy Manager. For more information, see the section on tuner and package properties in the *Policy Management User Guide*.

marimba.schedule.polite

specifies whether all updates initiated by the scheduler are done in trickle mode, with a maximum speed (in bytes per second) indicated by marimba.schedule.polite.speed. This property allows you to control the amount of bandwidth used by the tuner.

Valid value: true or false

Default value: false

marimba.schedule.polite.speed

specifies the speed (in bytes per second) at which updates occur if marimba.schedule.polite is set to true.

Valid value: an integer

Default value: 800 (bytes per second)

marimba.schedule.quietupdates

indicates whether the scheduler updates channels in a quiet mode. A scheduled (versus a manual) update, is usually quiet. No error or dialog boxes appear if problems occur during the update. Instead, this information is sent to the tuner's log file. If this property is set to false, scheduled updates occur in a non-quiet mode.

Valid value: true or false

Default value: true (scheduled updates are quiet)

marimba.schedule.restrict

is the minimum amount of time (in minutes) after which the scheduler performs updates. For example, to batch all scheduled updates into 20-minute intervals, set this property to 20.

`marimba.schedule.startdelay`

is the delay (in milliseconds) after the tuner starts and before the scheduler starts. This delay helps prevent the scheduler from monopolizing system resources when the tuner starts.

Valid value: an integer (milliseconds)

Default value: 120000 (2 minutes)

`marimba.security.cdrom`

indicates whether channels are able to access the CD-ROM drive on the local computer.

`marimba.security.cert.password.timeout`

is the number of seconds before client-certificate passwords time out. Specify -1 to indicate no timeout.

Default value: 3600 (1 hour)

`marimba.security.channels.onlytrusted`

indicates whether the tuner runs only channels that are trusted. Trusted channels are either signed with a trusted channel-signing certificate or are downloaded from a trusted transmitter. For more information about trusted transmitters, see `marimba.security.trusted.transmitters`.

Valid value: true or false

Default value: false (the tuner runs trusted and untrusted channels)

`marimba.security.clientcertpw`

sets the default base64-encoded password (for example, base64:bWFyaW1iYQ\=\=) for the client certificate used by the tuner.

`marimba.security.cookies`

indicates whether requests contain tuner IDs (also known as *cookies*).

`marimba.security.credentials.cache`

controls the behavior of Channel Copier with regard to storing credentials (user names and passwords). By default, if this property is set to `false`, Channel Copier does not store credentials in the session file. Channel Copier tries to use credentials in the tuner's keychain, or command line, if appropriate. If you want to change the behavior of Channel Copier (so that it stores credentials in the session file), set this property to `true`.

Valid value: true or false

Default: false

`marimba.security.identity.transmitters`

is a list of the transmitters that are informed of the user's login name and the tuner ID. This property has the same form as `marimba.security.trusted.transmitters`. If this property isn't set, its value is the same as that property.

`marimba.security.logging`

indicates whether requests contain log information.

`marimba.security.multicast.access`

indicates whether the tuner listens to and sends out multicast messages. See the tuner command-line options `-multicast` and `-nomulticast`.

`marimba.security.noUserOverride`

indicates whether users are presented with dialogs asking to override errors such as certificate expiration or unknown root certificates. Users are not presented with such dialogs if this property is set to true.

If this property is set to true, but the `marimba.security.trusted.certs` property is not defined, the tuner cannot start any signed channel, including Channel Manager.

Valid value: true or false

Default value: false

`marimba.security.printing`

indicates whether channels are allowed to use the printer.

`marimba.security.rpc.access`

indicates whether the tuner is able to open and receive RPC sessions. See the tuner command-line options `-rpc` and `-norpc`.

Valid value: true or false

`marimba.security.ssl.matchdomainonly`

indicates whether a tuner ignores the first part of an SSL certificate's common name when performing SSL certificate authentication. When a tuner connects to a transmitter, it normally verifies that the transmitter's host name matches the common name of the SSL certificate installed on the transmitter. When this property is set to true, the tuner matches only the domain part of both the transmitter's host name and the SSL certificate's common name.

For example, a machine `a.marimba.com` is running an SSL transmitter with a certificate with a common name of `b.marimba.com`. A tuner machine connecting to `a.marimba.com` through SSL usually shows a warning dialog explaining that the expected name (`a.marimba.com`) and the common name (`b.marimba.com`) on the certificate did not match. If this property is set to true, no such warning appears.

Use this property to facilitate administration of SSL transmitters behind a load balancer. When transmitters are running behind a load balancer, they typically have certificates installed on them with common names that do not match the host names of the transmitters.

Valid value: true or false

`marimba.security.sslcert`

sets the Certificate Manager's certID string (for example, `o4H7G0-Wf6FaW-0ljRib-pvsQ==`) for the tuner's default client certificate. If this property is set, the tuner always uses the corresponding certificate when client-certificate authentication is requested.

`marimba.security.trusted.certs`

is a list specifying which subject-issuer pairs the tuner always trusts. In the event that a certificate expires, a channel in this list still runs properly; however, updates do not work until the channel is republished with a valid certificate.

For each pair, the subject and issuer are separated by a vertical bar, as follows:

`<subject>|<issuer>`

Each subject and issuer pair is also separated by a vertical bar, as follows:

`<subject>|<issuer>|<subject>|<issuer>`

In order for a certificate to always be trusted, its subject and issuer must be a superset of some pair in this list. Either `<subject>` or `<issuer>` can be an asterisk (*), meaning any subject or issuer.

For example, the Marimba certificate matches any of the following subject-issuer pairs:

```
cn=Marimba, Inc.|ou=VeriSign Trust Network  
cn=Marimba, Inc.|*  
*|ou=VeriSign Trust Network  
*|*
```

See the tuner command-line option `-trust`.

`marimba.security.token.enable`

disables the password encryption feature of the tuner. You must set the `marimba.security.token.enable` property to false before you can use the `-anonymize` option. This property is set in the `prefs.txt` file.

`marimba.security.trusted.transmitters`

is a list of trusted transmitters. A channel that comes from a trusted transmitter is automatically granted the needed capabilities, without prompting the user. The value of this property is a list of items separated by semicolons; each item has one of the following forms:

- `<netmask>,<ip_addr>`, where `<netmask>` is used to find the significant bits in `<ip_addr>` to determine whether there is a match with a potential transmitter. If so, the matching transmitter is considered trusted. You can use `0` as a wild card for any portion of the netmask address. In the following example,

```
marimba.security.trusted.transmitters=
255.255.0.0,172.16.0.0;255.255.255.0,172.17.1.0;
255.255.0.0,192.168.0.0;255.255.255.255,127.0.0.1;
255.0.0.0,10.0.0.0
```

the first part (`255.255.0.0,172.16.0.0;`) means that all hosts with IP addresses that are `172.16.x.x` are trusted. Similarly, the netmask `255.0.0.0,10.0.0.0` means that all hosts with IP addresses that are `10.x.x.x` are trusted.

- `<DNS_host_name>`. The property allows trust to this host if it matches the host name exactly. This is useful in environments that require the use of the `marimba.proxy.trust` property, since these environments don't allow DNS resolution.

Note: This property is case-sensitive. For example, if the URL for the transmitter is `trans.acme.com`, then use the exact same case when specifying the transmitter for this property, not `TRANS.ACME.COM` or `Trans.Acme.com`.

`marimba.subscribe`

is a list of channels, separated by a vertical bar, which are subscribed to when the tuner starts. This property starts subscribing to channels 20 seconds after the tuner launches. Valid commands include start (optionally with channel-specific arguments), stop, subscribe, unsubscribe, update, and remove.

The syntax is `http://trans/Channel?<cmd><arg1>,<arg2>,<arg3>`

For example:

`marimba.subscribe=http://trans/Bin_Tree`

(subscribes to Bin_Tree)

`marimba.subscribe=`
`http://trans/Bin_Tree|http://staging/Subscription?start`

(subscribes to Bin_Tree, start Policy Service)

`marimba.subscribe=http://trans/PatchService?start-`
`install,debug,5`

(starts the Patch Service install with debug flag 5)

Note: If you have Policy Service installed, the tuner has additional properties that become available. You can set the following tuner properties (`marimba.subscription.*`) to configure the Policy Service, either when a tuner is packaged or through Policy Management.

marimba.subscription.adminusers

specifies, if the endpoint user's machine has Policy Service 5.0.1 or higher, the administrators who can log in to a user's machines temporarily when using user-based targeting (possibly for troubleshooting). This property prevents the user's channels from getting deleted when you log in as an administrator.

When the administrator logs in, the following occurs:

- Channels subscribed for another user are not removed.
- If channels have been assigned to an administrator, then these channels are delivered to the endpoint.
- The properties that have been set for the administrator are set. Possibly, the properties that have been set for the users are overwritten by those that are set for the administrator. However, when the users log back in and Policy Service updates, the users get their properties back.

Valid value: <user1>,<user2>,...

where <user1>,<user2>,... specifies a comma-delimited list of the administrators. The list of administrators can contain the name of any user who can log in at the endpoint machines.

Default value: none

marimba.subscription.dolast

specifies the URL of the channel that is applied last as part of a policy. This property is only available with Policy Management (previously called Subscription) 4.6.x and 4.7.x.

marimba.subscription.installmode

defines the mode used by Policy Service while installing or uninstalling channels.

If you want a completely silent installation, you can set these tuner properties to prevent progress bars or error dialog boxes from appearing on endpoints: marimba.tuner.display.noprogress and marimba.tuner.display.noerrors.

Valid value:

- **silent**—Policy Service installs all packages on the target tuner in silent mode.
- **aspackaged**—Policy Service uses the installation mode specified in the package.

Default value: silent

`marimba.subscription.inventory.compliance`

enables the collection of compliance data for the endpoint. By default, the collection of compliance data from endpoints is disabled. To set this property for all endpoints (that is, to enable the collection of compliance data for all endpoints), use the Enable collection of compliance data option on the Configuration > Compliance options page of Policy Manager.

Valid value: true or false

Default: false

`marimba.subscription.machinename`

allows you to override the machine name returned by Policy Service. The name you enter in this property must match the name used in the machine's flat file. By default, this property has no value, and the machine name is the DNS host name without the domain information.

Valid value: a string representing the name of the machine

Default value: none

`marimba.subscription.nodelete`

specifies whether packages are deleted. By default, packages are deleted when a user or machine is removed from a group in the underlying directory server. If you set this property to true, packages are not deleted even if you delete packages from targets using the Target Details page. Packages remain on the targets and are no longer managed using Policy Management.

Valid value: true or false

Default value: false

`marimba.subscription.reapplyconfigonfail`

determines whether to reapply a cached policy if Policy Service cannot communicate with the plug-in to get an updated policy. This property is available in version 6.0 and higher.

When set to `true`, Policy Service reapplies the cached policy (the last policy downloaded) if it cannot communicate with the plug-in to get an updated policy. When set to `false`, Policy Service does not attempt to reapply the cached policy.

Valid value: `true` or `false`

Default value: `true`

`marimba.subscription.reboot.allowcancel`

specifies whether the Cancel button is displayed. By default, the Policy Service Reboot dialog box displays a Cancel button along with a Reboot Now button. When this property is set to `false`, the Cancel button is not displayed.

Valid value: `true` or `false`

Default value: `true`

`marimba.subscription.retrycount`

sets the number of retries before Policy Service stops trying.

Valid value: an integer

Default value: 5

`marimba.subscription.retryintervalsec`

sets the interval to wait, in seconds, before retrying the connection to the directory service. You set this property on the tuner hosting the transmitter on which you published the Policy Service plug-in.

Valid value: an integer

Default value: 30 (30 seconds)

`marimba.subscription.retrytime`

sets the delay, in seconds, before Policy Service tries to subscribe failed channels.

Valid value: an integer

Default value: 60 (60 seconds or 1 minute)

marimba.subscription.timeout

sets the maximum period of time, in seconds, that Policy Service waits for an operation to complete (such as a package sending a notification that it has achieved a specified state) before proceeding.

Valid value: an integer

Default value: 3600 (3600 seconds or 1 hour)

marimba.subscription.ucd.enabled

enables User Centric Deployment feature on an endpoint.

Valid value: true or false

Default value: false

marimba.subscription.updatealways

no longer supported; always true.

marimba.subscription.update.schedule

allows the Policy Service update schedule to be set when a tuner is packaged. When this property is set, it is applied each time Policy Service starts.

Valid value: schdeule string

For example: marimba.subscription.update.schedule=every 2 days update at 4:00AM

For more information on schedule strings, see Chapter 5, “Syntax for the schedule string.”.

If you set the value of this property to null, the update.schedule property is removed from the policy service channel.txt file.

marimba.subscription.usecomputername

specifies whether Policy Management uses the Windows NetBIOS name. By default, Policy Management uses the DNS or TCP/IP host name as the machine name. When this property is set to true, Policy Management uses the Windows NetBIOS name instead.

Valid value: true or false

Default value: false

marimba.subscription.useshortcuts

specifies whether advertised packages (available channels) are represented as desktop shortcuts. This property is used for Windows platforms only.

Valid value: true or false

Default value: false

marimba.subscription.varytime

sets the maximum period of time, in minutes, that scheduled events (such as downloading or updating a package) can be postponed. This setting improves transmitter performance by spreading out endpoint requests during periods of heavy load. For example, if this property is set to 10 minutes and an event is scheduled to occur at 10:00 am, then the event can occur any time between 10:00 am and 10:10 am.

Valid value: an integer

Default: 10 (10 minutes)

marimba.subscription.usexml

specifies whether advertised packages (available channels) try to resolve package titles and categories using the transmitter's XML listing.

Advertised packages do not try to resolve titles and categories when this property is set to false.

Valid value: true or false

Default: true

marimba.tuner.admin

specifies who is allowed to remotely administer the tuner. The property value can be any of the following:

- empty string (no one)
- * (anyone)
- <user_name>,plain:<password>
- auth:<user_name> (authenticated user)
- auth:group=<groupID>[;<groupID2>;...] or
auth:group=<groupDN>[;<groupDN2>;...]
- auth:*(any authenticated user)

For more information, see the tuner command-line option -admin (in “Tuner options” on page 27.

If you are using a directory service (LDAP) for authentication, you can use the following form:

```
marimba.tuner.admin=auth:{*|<userID>|group=<groupID>[;<groupID>2;...]|group=<groupDN>[;<groupDN>2;...]}
```

where:

- auth:***—Any authenticated user can connect to the tuner.
- auth:<*userID*>—An authenticated user with this specific user ID can connect to the tuner. The user ID must be unique within the directory service (LDAP).
- auth:group=<*groupID*>[;<*groupID2*>;...] or
auth:group=<*groupDN*>[;<*groupDN2*>;...]—An authenticated user from the specified group or groups can connect to the tuner. If you specify multiple groups, use semicolons to delimit the list. You can specify group IDs or DNs. The group DN syntax is supported only when marimba.tuner.admin.ldap.useouforgroups={true|false} is set to false. Nested groups are not supported.

For example:

- marimba.tuner.admin=auth:spiderman
- marimba.tuner.admin=auth:group=ops
- marimba.tuner.admin=auth:
group=cn=ops,ou=groups,o=marimba.com;
cn=backups,ou=groups,o=marimba.com

```
marimba.tuner.admin.emergency
```

specifies an emergency password that allows you or other users to connect to the tuner even when the tuner cannot connect to the directory service (LDAP). This password is optional. If specified, it must have the following form:

```
marimba.tuner.admin.emergency=MD5:<digest>
```

where <*digest*> is the MD5 digest of *:*password*. (MD5 is case-sensitive.)

For example, to specify an emergency password of opensesame:

```
marimba.tuner.admin.emergency=MD5:Rlrf5WmPren/2VHAd0F/A==  
marimba.tuner.admin.ldap.basedn=<dn>
```

specifies the base distinguished name (DN) to use when performing directory service (LDAP) searches. A null base DN is not allowed.

For example: `marimba.tuner.admin.ldap.basedn=dc\=company,dc\=com`
`marimba.tuner.admin.ldap.binddn=<relative_dn>`

specifies the bind DN to use when creating the initial directory context. You are required to set this property if anonymous binds are not permitted for the directory service (LDAP). Specify the bind DN as a relative DN (relative to the base DN).

For example:

`marimba.tuner.admin.ldap.binddn=cn\=BindDNService,\ou=users,`
`dc\=company,dc\=com`

`marimba.tuner.admin.ldap.countlimit=<max>`

specifies the maximum number of entries that are returned by directory service queries.

Default value: 500

`marimba.tuner.admin.ldap.enable={true|false}`

enables the tuner to use a directory service (LDAP) for authentication when set to true. If the property is undefined or set to false, the tuner ignores all the tuner properties related to directory service authentication.

Valid value: true or false

`marimba.tuner.admin.ldap.groupclass=<gc>`

specifies a comma-delimited list of object classes used to represent a group. This property is valid only if

`marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to false.

Default value: group, groupOfNames, groupOfUniqueNames

`marimba.tuner.admin.ldap.groupexcludeclass=<gc>`

specifies a comma-delimited list of group object classes to exclude when performing group searches. This property is valid only if

`marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to false.

Valid value: groupOfNames, groupOfUniqueNames

`marimba.tuner.admin.ldap.groupmemberattr=<gm>`

specifies the attribute used to determine the members of a group. This property is valid only if

`marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to false.

Default value: member, uniquemember

`marimba.tuner.admin.ldap.groupnameattr=<gn>`

specifies the attribute used to identify a group. This property is valid only if `marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to false.

Default value: cn

`marimba.tuner.admin.ldap.onlyreturnusers={true|false}`

specifies whether to return only objects with the user ID attribute set. If this property is set to false, then non-user group members are returned from searches. This property is valid only if

`marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to false.

Valid value: true or false

Default value: false

`marimba.tuner.admin.ldap.ouattr=<oa>`

specifies the attribute used to determine the groups to which an object belongs. This property is valid only if

`marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to true.

Default value: ou

`marimba.tuner.admin.ldap.ouclass=<oc>`

specifies the object class used to represent a group. This property is valid only if `marimba.tuner.admin.ldap.useouforgroups={true|false}` is set to true.

Default value: organizationalUnit

`marimba.tuner.admin.ldap.password=<b64password>`

specifies the base64-encoded bind password to use when creating the initial directory context. You are required to set this property if anonymous binds are not permitted for the directory service (LDAP).

For example, to specify the password as opensesame,
marimba.tuner.admin.ldap.password=b3B1bnNlc2FtZSANCg==
marimba.tuner.admin.ldap.searchdns={true|false}
specifies whether to search for group attributes in object DNs when
performing group searches. This property is valid only if
marimba.tuner.admin.ldap.useouforgroups={true|false} is set to
true.

Valid value: true or false

Default value: false

```
marimba.tuner.admin.ldap.server=<host1>:<port1>,[<host2>:<port2>,...]
```

specifies the host and port number for one or more directory services (LDAP). If you specify more than one directory service, use a comma to delimit the items in the list. The tuner tries to use the first available directory service, and moves down the list of services if necessary. If a server determines that the user name and password are invalid, the tuner does not try the other servers in the list. If the user name and password are invalid, or if no servers are available, then users are not able to connect to the tuner unless they use the emergency password. (See `marimba.tuner.admin.emergency`.)

For example:

```
marimba.tuner.admin.ldap.server=primary:389,secondary:389
```

```
marimba.tuner.admin.ldap.ssl={true|false}
```

specifies whether SSL is used when connecting to the directory service (LDAP). If this property is set to `false`, simple authentication sends passwords in clear text to the directory service.

Valid value: `true` or `false`

Default value: `false`

```
marimba.tuner.admin.ldap.type={iplanet|msadnative|msadmixed}
```

specifies the directory service (LDAP) schema to use.

If you use mixed mode (`msadmixed`), the user name you enter is in the form `<name>` (for example, `simon`). If you use native mode (`msadnative`), the user name you enter is in the UPN form `<name>@<domain_name>.com` (for example, `simon@marimba.com`).

The properties from

```
marimba.tuner.admin.ldap.countlimit=<max> to  
marimba.tuner.admin.ldap.useridattr=<uid>
```

allow you to configure advanced directory service (LDAP) settings. You set these properties only if the directory service you are using is customized for your organization and does not use the default settings for the directory service schema (iPlanet or Active Directory).

Valid value:

- `iplanet`—The Netscape iPlanet or Oracle Directory Server
- `msadnative`—The Microsoft Active Directory in native mode (all Windows 2000)

- msadmixed—The Microsoft Active Directory in mixed mode (Windows NT/NetBios)

Default value: iPlanet schema

```
marimba.tuner.admin.ldap.useouforgroups={true|false}
```

specifies whether an organizational unit containment model is used. If set to true, each user has one or more organizational unit attributes that identifies the group or groups to which the user belongs. If set to false, the group of names containment model is used, and each group has member attributes identifying each user that belongs to the group.

Valid value: true or false

Default value: false

```
marimba.tuner.admin.ldap.userclass=<uc>
```

specifies the object class used to represent a person.

For example: marimba.tuner.admin.ldap.userclass=user

Default value: inetorgperson

```
marimba.tuner.admin.ldap.useridattr=<uid>
```

specifies the attribute used to identify a user.

For example: marimba.tuner.admin.ldap.useridattr=cn

Default value: uid

```
marimba.tuner.deprecate64bitsegment.channels
```

specifies the list of custom channels that contain DLLs for 32-bit or 64-bit platforms. For these channels, if there are any 64-bit segments then it will be deprecated. So for these channels, the tuner will always bring down 32-bit segment. The custom channels included in the list should be separated by a comma.

For example:

```
marimba.tuner.deprecate64bitsegment.channels=http://localhost:5282/Tools/CustomChannel,http://localhost:5282/Tools/CustomChannel1
```

If you want to deprecate a 64-bit segment for custom channels then you can set this property on the Windows machine for all profiles except the 64-bit master, mirror and repeater profile. This ensures that the tuner brings down 32-bit segment of the custom channel instead of 64-bit segment.

`marimba.tuner.display.nocancel`

indicates whether the tuner shows a Cancel button in the progress indicator box that appears when installing or updating a channel. Set this property to `true` if you do not want the Cancel button to appear.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.display.noerrors`

indicates whether the tuner shows error and warning dialogs. To hide error and warning dialogs, set this property to `true`. The error and warning messages are printed out to a system console. This property does not apply if you are running the tuner without a display.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.display.noprogress`

indicates whether the tuner shows a progress bar when channels are being subscribed to or being updated. To hide progress bars, set this property to `true`. This property does not apply if you are running the tuner without a display.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.display.nosavedpasswords`

indicates whether the user's password is saved in the tuner's keychain. Using a Tuner version 4.6.1.19 or higher, if you set this property to `true` and use Channel Copier (or Tuner Administrator, Transmitter Administrator, Publisher, and so on) of any version, then the Don't Ask Again check box is not displayed. Therefore, credentials that users enter are not saved in the tuner's keychain.

Default value: `false`

`marimba.tuner.display.nowarnings`

indicates whether the tuner shows warning dialogs. To hide warning dialogs set this property to `true`. The warning messages are printed out to a system console. This property does not apply if you are running the tuner without a display.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.display.version`

specifies the tuner version you want displayed in the following:

- the progress bar
- the name of the uninstall DLL file
- the tooltip for the tuner icon that appears in the taskbar

Set this property to a numeric value (for example, 1.3 or 4.6). The property is useful in branded tuners, where the tuner version is inappropriate. For example, the tuner version can be 4.6 but the customer's product is version 1.2.

`marimba.tuner.enabletaskbaricons`

suppresses the display of any task bar icons (such as Windows) on platforms that support such icons. Set this property to `false` to suppress the display. This property applies to the tuner's icon and channel icons. It overrides properties for icon display that can exist at the channel level. Currently, the proxy and transmitter support taskbar icon display through their own channel properties (`proxy.enableTaskBarIcons`, and `main.enableTaskbarIcons` respectively).

Valid value: `true` or `false`

Default value: `true` (since 4.6.1)

`marimba.tuner.internetdeploy.enabled`

enables Marimba over Internet feature. If this property is not set, or set to false, this feature is disabled. By default, this feature is disabled.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.iphost.expire`

specifies the time (in minutes) after which the IP address associated with a channel expire. This IP address comes from the transmitter from which a channel was downloaded. The expire policy is checked, and accordingly, the IP address can refresh the next time this channel updates. The expire policy can be one of the following:

- -1 indicates that the IP address never expires. The tuner does not try to re-resolve the IP address even if the cached IP address does not work. This value is not recommended if you want the tuner to re-resolve the IP address if the cached IP address does not work.
- 0 indicates that the IP address for the channel is not kept in the cache. Each time the channel is updated, the DNS lookup is repeated.
- <min> indicates that the IP address expires after the specified number of minutes. The next time the channel is updated, the DNS lookup is repeated *only if* the current IP address has expired according to the <min> setting.

When you set this property, you are setting the expire policy for all the channels on that tuner. So, the next time any channel on that tuner is updated, the IP address associated with that channel is also updated if it has expired.

You can also choose to update the IP addresses associated with some channels (but not other channels) on a tuner by setting the `iphost.expire` property for those channels. This property overrides `marimba.tuner.iphost.expire`. For example, you can set `marimba.tuner.iphost.expire` to -1, and set the `iphost.expire` property of channels A, B, and C to 0. The IP addresses associated with *only* channels A, B, and C are updated the next time these channels update.

The default value for the `marimba.tuner.iphost.expire` property on JRE 1.1.8 is -1, which indicates that the IP address associated with a channel never expires. Starting in the 4.6.2.1 release, the default value for this property on JRE 1.3.1 is 10, which indicates that the IP address expires after 10 minutes. Because `marimba.tuner.iphost.expire` is a tuner launch property, restart the tuner after changing its value.

If you previously set the Sun Microsystems property `sun.net.inetaddr.ttl` as a workaround using any of these methods:

- Setting the tuner properties `marimba.launch.javaArgs` or `marimba.launch.NTServiceArgs`
- Using the command-line option `-java <java_arg>`

then the following applies:

`marimba.launch.javaArgs` or `marimba.launch.NTServiceArgs` takes precedence over `marimba.tuner.iphost.expire`. If the `marimba.launch.javaArgs` or `marimba.launch.NTServiceArgs` property is set, then the `marimba.tuner.iphost.expire` value is not honored.

If `marimba.launch.javaArgs`, `marimba.launch.NTServiceArgs`, and `marimba.tuner.iphost.expire` are not set, then the default cache refresh rate is 10 minutes.

If `marimba.launch.javaArgs` is not set and `marimba.tuner.iphost.expire` is set, the `sun.net.inetaddr.ttl` is set to the value of `marimba.tuner.iphost.expire`.

`marimba.tuner.iphost.refresh`

indicates whether to refresh the tuner's address cache at startup. The address cache caches the DNS name and corresponding IP address of the source transmitter from which a channel is downloaded. If this property is set to `true`, the tuner checks the expire policy (specified by `marimba.tuner.iphost.expire`) at startup and refreshes the IP addresses of all channels on the tuner if necessary. If this property is set to `false`, the tuner does not check the expire policy at startup and does not refresh any IP addresses.

For example, if `marimba.tuner.iphost.expire` is set to 0 (indicating that the IP address always expires) and `marimba.tuner.iphost.refresh` is set to `true`, then at startup, the tuner refreshes the IP addresses for all channels. Similarly, the tuner refreshes the IP address at startup if `marimba.tuner.iphost.refresh` is set to `true` and the number of minutes specified in the `marimba.tuner.iphost.expire` property's `<min>` value have passed. On the other hand, if `marimba.tuner.iphost.expire` is set to -1 (indicating that the IP address never expires), and the `marimba.tuner.iphost.refresh` is set to `true`, then at startup the tuner still does not refresh the IP addresses for all channels.

You can choose to refresh the IP addresses of some channels (but not others) on a tuner by setting that channel's `iphost.expire` property. This channel property overrides the tuner's `marimba.tuner.iphost.expire` property. For example, you can set `marimba.tuner.iphost.expire` to -1, and set the `iphost.expire` property of channels A, B, and C to 0. If `marimba.tuner.iphost.refresh` is set to `true`, the next time the tuner starts, it refreshes the IP addresses associated with only channels A, B, and C.

This property was added in version 4.6.1.13.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.keeprunning`

indicates whether the tuner keeps running after the last channel (typically Channel Manager) stops running. By default, Windows tuners (with taskbars) keep running.

Starting with Tuner version 4.6.2.2., if the platform is other than Windows, the tuner properties `marimba.tuner.keeprunning=true` and `marimba.tuner.minimize=false` are automatically set. This change keeps users from having to make these changes during tuner packaging to make sure that the installed tuner does not exit if no channels are running.

See the `service.daemon.channel` property as described in the *Developing Marimba Channels*.

Valid value: true or false

`marimba.tuner.logs.applyFilters`

tells the Logging Service channel to apply filters to the log messages produced on the endpoint. The filtered messages are collected in a special log and then sent back to the Logging plug-in on the transmitter according to a schedule and when a triggering event occurs. You use Report Center to specify the triggering events.

You set this property to `false` on an endpoint if you want to disable the filtering and just send every log message back to the database. You can do this in troubleshooting situations, for a short time, and then turn the filtering back on.

Valid value: true or false

Default value: true

`marimba.tuner.logs.centralizedlogging`

enables the Logging Service channel to run when the tuner starts.

WARNING: By default, this property is set to `false`. If you set it to `true` to use centralized logging, all log messages of the severity level `MAJOR` are collected for every channel on the endpoint. (You can later use Report Center to change this default configuration.) Collecting all these log messages can result in a large volume of data inserted into your database, depending on how many endpoints you have. Therefore, it is recommended that you leave this property set to `false`, and then, after you use Report Center to limit the log ID ranges collected, you can use either Policy Manager or Deployment Manager to set this tuner property to `true` on endpoints.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.logs.platform.eventsource`

determines the event source for the log event. The value of the property represents the application from which the event came.

Valid value: a string

Default value: `Marimba Tuner`

`marimba.tuner.logs.platformlogging`

turns platform-specific logging on or off for the entire tuner. Platform-specific log messages go to the operating system's logging facility (for example, the event logs on Windows). Currently, this property is supported only on Windows NT, Windows 2000, and Windows XP, mainly for use with the tuner as a service. When set to `false`, no logs are written to the Windows Event logs from any source, such as java tuner, minituner, or tuner launcher.

Note: Use this property with caution. Turning on logging for the tuner using this property can greatly increase the needed size of the event log (caused by numerous log messages from the tuner).

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.logs.platform.logname`

determines the system log in which the entries are sent. Currently, this property is applicable only under Windows NT, Windows 2000, and Windows XP, which have the concept of multiple logs. When this property is not set, the default is the Application log, which is generally the right option. You should rarely set this property.

Valid value: a string

Default value: Application

`marimba.tuner.logs.roll.policy`

specifies the policy for rolling the tuner's log files.

If you choose `manually` or `never`, the log entries are recorded in a single file, which is never automatically rolled. If you choose `bysize`, the log file is rolled automatically when it reaches the size specified in `marimba.tuner.logs.roll.size`.

Note: Due to a current limitation, if you set `marimba.tuner.logs.roll.size` to anything other than the default (128 kilobytes), then `marimba.tuner.logs.roll.policy=bysize` is no longer the default. Set the property explicitly.

Valid value: hourly, daily, weekly, monthly, yearly, manually, never, `bysize`

Default value: `bysize`

`marimba.tuner.logs.roll.size`

specifies the size in kilobytes the tuner's log file must reach before the file is rolled automatically. The value of this property is used when `marimba.tuner.logs.roll.policy` is set to `bysize`.

Default value: 128

`marimba.tuner.logs.roll.versions`

specifies the number of previously rolled log files for the tuner that can exist.

Default value: 1

`marimba.tuner.minimize`

prevents the tuner from being minimized when set to `false`.

Starting with Tuner version 4.6.2.2., if the platform is other than Windows, the tuner properties `marimba.tuner.keeprunning=true` and `marimba.tuner.minimize=false` are automatically set. This change keeps users from having to make these changes during tuner packaging to make sure that the installed tuner does not exit if no channels are running.

Valid value: true or false

Default value: true

`marimba.tuner.minimize.log`

specifies whether the tuner logs transitions when entering and exiting minimal mode. If set to true, log entries are written to the tuner's log. The reason for exiting minimal mode is also included in the log entry. This property is applicable only when `marimba.tuner.minimize` is set to true.

Valid value: true or false

Default value: false

`marimba.tuner.name`

is the human-readable name of the tuner. This is primarily used by Policy Management, and can be set to the local IP address.

Valid value: a string

Default: none

`marimba.tuner.p2p.bcast.addr`

indicates the broadcast address used by the tuner to search for peers via UDP along with the UDP port number. For more information on MESH, see the *Symphony Marimba Client Automation CMS and Tuner User Guide*.

`marimba.tuner.p2p.enabled`

specifies whether MESH mode is enabled or not. For more information on MESH, see the *Symphony Marimba Client Automation CMS and Tuner User Guide*.

`marimba.tuner.p2p.filepeerphase.disabled`

enables or disables the File Peer Phase. If this property is set to true, then the tuner skips the File Peer Phase even when there are pending files after Channel Peer Phase. The tuner communicates directly with the Transmitter for downloading the required files. If you set this property is to false, File Peer Phase is enabled. By default, the property is not set and hence the default value is false.

Valid value: true or false

Default value: false

`marimba.tuner.p2p.port`

specifies the listening port on which tuners connect to receive files. For more information on MESH, see the *Symphony Marimba Client Automation CMS and Tuner User Guide*.

This property is deprecated.

`marimba.tuner.p2p.udp.waittime`

This property is used by the requesting peer tuner to determine how long it should wait after sending broadcast packets for the required files, before it stops accepting replies from peer tuners which are ready to serve files. In case of scenarios where a large number of packages are getting subscribed or updated on an endpoint tuner at the same time, a high value of this property results in less probability of packet loss. By default, this property is not set and when not set, the default value is 15 seconds.

`marimba.tuner.network.httpTimeout`

specifies the idle timeout for HTTP requests. After a successful connection, HTTP requests end connections after the idle timeout period expires. If this property is set to 0, then the connection never times out.

Valid value: an integer

Default value: 90 (90 seconds)

`marimba.tuner.nt.reflect.username`

specifies whether the tuner includes the user name stored in `runtime.tuner.nt.username` when it is communicating with the transmitter.

Knowing the user name a tuner sends is important, for example, when you are using a user name-based policy. The user name included in an update request is retrieved from the `user.name` system property. The tuner sets this system property from one of the properties described below:

- `runtime.os.user`—This property is a Java system property that identifies the user who owns the executing JVM process. By default, the tuner uses the user name stored in this property. Regard this property as read-only.

- `runtime.tuner.nt.username`—On Windows NT, this property is automatically updated with the user name of the user who is logged on to the machine. This property is useful, for example, when a service has the `runtime.os.user` property set to SYSTEM. The value of the `runtime.tuner.nt.username` property is used only if `marimba.tuner.nt.reflect.username` is set to true. This property is automatically set by your operating system. Regard this property as read-only.
- `marimba.tuner.user`—This property can be set manually or programatically (for example, by creating a channel that sets this property on endpoints). This property is used when it is non-null and when `marimba.tuner.nt.reflect.username` is false. You use the `marimba.tuner.user` property when you don't want the user name to be determined by who is currently logged on, or by the user who owns the executing JVM process.

The following steps show the user name selection process:

1. If you are on Windows NT *and* if `marimba.tuner.nt.reflect.username` is true, then the user name stored in `runtime.tuner.nt.username` is used.
2. Otherwise, if `marimba.tuner.user` is specified, then this user name is used.
3. Otherwise, the user name stored in `runtime.os.user` is used.

Valid value: true or false

Default value: true

`marimba.tuner.nt.username.persist`

specifies whether `runtime.tuner.nt.username` is set to `SYSTEM` when the user logs off. When the user logs off, the tuner checks for `marimba.tuner.nt.username.persist`. If this property is set to `true`, then instead of setting `runtime.tuner.nt.username` to `SYSTEM`, `runtime.tuner.nt.username` is set to the last logged in user name. If a scheduled update occurs, the last logged in user name is sent to the Policy Service plug-in and subsequently all the user's channels are not deleted in the tuner's workspace. When the user logs back in, `runtime.tuner.nt.username` is set to the logged in user name again.

This property takes effect only if the

`marimba.tuner.nt.reflect.username` property is set to `true`.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.outgoing.host`

is the default outgoing IP address used by the tuner's HTTP environment interface. Most non-RPC outgoing tuner connections use this value as the outgoing address. This address property is useful when running the tuner on a multi-homed system that has multiple IP addresses (real or virtual).

`marimba.tuner.receipt.maxdays`

specifies the maximum number of days receipts stay in the tuner before purging.

Valid value: integer

Default value: 365 days

`marimba.tuner.receipt.maxsize`

specifies the maximum size in MB of a receipt.

Valid value: integer

Default value: 300 MB

`marimba.tuner.release.version`

is the version identifier for the tuner. This value is set during the build process, and it has the following form:

`<major_version>.<minor_version>.<micro_version> <modifier>`

For example: 3.0.0 beta 6

If you use the Channel Manager, Channel Manager compares this property with an internal value to see if the tuner must be updated.

Valid value: a string

`marimba.tuner.restart.timeout`

specifies the maximum number of seconds that the tuner waits for all uninterruptible channels to stop before restarting the tuner.

If a channel attempts to restart the tuner, the tuner detects if any uninterruptible channels are running. Uninterruptible channels have the channel property `interrupt` set to `false` and are usually the service channels (Deployment Service, Infrastructure Service, Patch Service, Policy Service, and Scanner Service). The tuner delays the restart until all uninterruptible channels stop. If one or more uninterruptible channels become unresponsive (or do not stop), the tuner restarts after the number of seconds specified by this property.

Valid value: an integer (time in seconds)

Default value: 3600 (1 hour)

`marimba.tuner.rpc.cert.id`

is the certID or unique ID that is assigned by the current certificate manager for the certificate used by RPC over an SSL connection.

`marimba.tuner.rpc.host`

is the host interface name that RPC starts listening on. This host is different from `runtime.rpc.host`, which is the host that RPC is actually running on.

This property takes effect only if `marimba.tuner.rpc.port` or `marimba.rpc.sslport` is set. This property is ignored if RPC options are supplied on the tuner command line.

`marimba.tuner.rpc.pool`

is the number of users that can access the same Tuner Administration browser at the same time.

The general rule is that one user uses 5 RPC threads, so you should set the thread pool size at five times the number of anticipated users. For example, if you expect 10 users, set the property to 50.

`marimba.tuner.rpc.port`

is the port that RPC starts on. This port is different from `runtime.rpc.port`, which is the port that RPC is actually running on.

This property is ignored if RPC port arguments are supplied through the command line.

Default value: 7717

`marimba.tuner.rpc.secure`

specifies whether an SSL RPC listener (instead of plain text) is used. This property requires that `marimba.tuner.rpc.certpw` be set or that the tuner be started with the `-certPassword` argument. Also, specify the SSL certificate in `marimba.tuner.rpc.cert.id`.

Default value: false

`marimba.tuner.rpc.ssllisten` (deprecated)

`marimba.tuner.rpc.sslport` (deprecated)

is the port that RPC or SSL RPC (respectively) starts on, or null if this feature isn't needed. This port is different from `runtime.rpc.sslport`, which is the port that RPC or SSL RPC is actually running on.

This property takes effect only if `marimba.tuner.rpc.host` is set.

`marimba.tuner.sessionisolation.enabled`

enables session isolation on the endpoints. To enable Session isolation, set the `marimba.tuner.sessionisolation.enabled` property to true and restart the tuner. Once the tuner restarts, the tuner and Java executables launch in Session 0 and Marimba Client starts in the current logged-on user session.

Valid values: true or false

Default value: false

`marimba.tuner.startmenu`

If you set this property to true, the primary channel is launched after the tuner startup or when the tuner wakes up from minimal mode.

BBCA recommends that this property should be set when the channel manager is configured as primary URL.

Valid value: true or false

`marimba.tuner.status.enable`

enables or disables tuner diagnostics. The default is enabled (true).

Valid value: true or false

`marimba.tuner.socket.buffersize`

specifies the size of the operating system packet queue that is used by the UDP socket for queuing up MESH UDP packets. If you specify a high value to this property, the possibility of packet loss in the UDP broadcast phase decreases. By default, this property is not set and the default value is 1048576.

You can use this property to set the length of the TCP packet queue in the following locations:

- The receiving buffer size of the RPC port at which the tuner listens for requests. If this property is not set, the buffer size is by default set to 1024000.
- The sending buffer size when a tuner initiates a HTTP connection. For example, you can specify the size of the buffer for a tuner which connects to a Transmitter for updates or subscribes. If this property is not set, the buffer size is by default set to 1024000. The default value ensures that the transfer of packets using the RPC port of the tuner utilizes maximum bandwidth to minimize packet loss. For example, when the tuner works as a peer in MESH mode.

`marimba.tuner.trayicon.menu.about.enabled`

removes the About menu item from the system tray icon for the tuner when set to false.

Valid value: true or false

`marimba.tuner.trayicon.menu.cancel.enabled`

removes the Cancel menu item from the system tray icon for the tuner when set to false. The Cancel menu item does not appear if the tuner is running as a service with no display, even if this property is set to true.

Valid value: true or false

`marimba.tuner.trayicon.menu.exit.enabled`

removes the Exit menu item from the system tray icon for the tuner when set to `false`. The Exit menu item does not appear if the tuner is running as a service with no display, even if this property is set to `true`.

Valid value: true or false

`marimba.tuner.trayicon.menu.open.enabled`

removes the Open menu item from the system tray icon for the tuner when set to `false`. The Open menu item does not appear if the tuner is running as a service with no display, even if this property is set to `true`.

Valid value: true or false

`marimba.tuner.trayicon.tooltip.normal.text`

modifies the tooltip displayed when the cursor is on the system tray icon for the tuner and when the tuner is *not* receiving data from the network.

Valid value: a string

`marimba.tuner.trayicon.tooltip.receiving.text`

modifies the tooltip displayed when the cursor is on the system tray icon for the tuner and when the tuner is receiving data from the network.

Valid value: a string

`marimba.tuner.update.credentials`

specifies the default credentials to use for a `libchannel` update. These credentials are a Base64-encoded string of the following form:

`<user_name>:<password>`

`marimba.tuner.uninstall.allowed`

allows an endpoint user to uninstall the tuner from Add or Remove Programs if `marimba.tuner.uninstall.allowed=true`. The default value is `false`.

`marimba.tuner.update.name`

specifies, if kernel updates must be signed, the distinguished name that must match the certificate that the update is signed with.

`marimba.tuner.update.periodic`

indicates whether Tuner Update Manager checks for updates periodically according to its channel `start.schedule` property. Tuner Update Manager is started periodically regardless of this property setting, since the `start.schedule` property is set at publish time. However, Tuner Update Manager does not check for an update unless `marimba.tuner.update.periodic` is true.

Note: This property has been deprecated.

`marimba.tuner.update.periodic.silent`

determines the user interface that is presented to the user during periodic tuner kernel updates.

Valid value:

- `false`—Fully interactive updates.
- `true`—No user interaction during updates. Users are not asked if they want to update, and the tuner is restarted if necessary. If an error occurs, the update fails silently. This value is useful for tuners set to run as a service on Windows or as a server application on UNIX servers.
- `displayonly`—An interface displays during updates, but the user is not required to click any buttons. This value is useful for kiosks and demos.

Default value: `false`

Note: This property has been deprecated.

`marimba.tuner.update.profile`

specifies the name of the profile segment to apply to a tuner during an update. The name of the profile segment typically takes the form `.profile_<profile_name>`. The absence of this property causes Infrastructure Service to apply only binary upgrades. This property was added for 6.0.

Valid value: a string representing a profile segment (typically with the form `.profile_<profile_name>`)

`marimba.tuner.update.restart`

specifies whether Infrastructure Service restarts the tuner after an upgrade. The absence of this property is the same as setting it to true. This property was added for 6.0.

Valid value: true or false

Default value: true

`marimba.tuner.update.schedule`

specifies the execution schedule for Infrastructure Service. This schedule is applied to Infrastructure Service configuration by Infrastructure Service itself. This property should match the form of the channel property `start.schedule`. This property was added for 6.0.

Valid value: See “Syntax for the schedule string” on page 259.

`marimba.tuner.update.seturl (deprecated)`

indicates whether a `setURL()` operation is performed on the primary channel after a successful kernel update.

`marimba.tuner.update.unsigned`

indicates whether kernel updates must be unsigned.

`marimba.tuner.url`

is the URL of the tuner used by the Tuner Update Manager to update the tuner kernel. See the tuner command-line option `-updateFrom`.

`marimba.tuner.user`

is the user name of the current tuner user. Channels can set this property, causing the tuner to set the operating system property `user.name`. This property is saved to disk.

`marimba.tuner.workspace.dir`

is the tuner’s workspace directory. The actual name on disk can have the RPC port number appended, as described for the tuner command-line options. See the tuner command-line option `-rpc`.

`marimba.tuner.p2p.allowfileneeded`

specifies whether the tuners in minimal mode can respond to the requesting tuners during the File Peer phase of MESH. If you set the `marimba.tuner.p2p.allowfileneeded` property to `true`, the tuners in minimal mode respond to the requesting tuners during the File Peer phase of MESH. If this property is set to `false`, the tuners in minimal mode do not respond to the requesting tuners during the File Peer phase of MESH.

The default value of this property is set to `false` to prevent tuners in minimal mode from responding to requesting tuners during the File Peer phase of MESH. However, irrespective of the value of this tuner property, the tuners in minimal mode always respond to requesting tuners during the Channel Peer phase of MESH.

Valid value: `true` or `false`

Default value: `false`

`marimba.tuner.network.ip.checkdelay`

specifies whether the tuner monitors the changes in the computer's IP address. The value of this property specifies the interval at which the tuner checks whether the IP address of the computer has changed. If you set the value of this property to -1, the tuner does not monitor the changes in the IP address of the computer.

Valid value: seconds

Default value: 30

`marimba.tuner.p2p.nopeers`

specifies the number of peers that a tuner connects to other peers during the Channel Peer phase of MESH. This property limits the total time which a MESH-enabled subscribe or update requires by limiting the number of peers the tuner connects to other peers for files. You can use this property to ensure that the load is not distributed across many peers. If you do not specify this property the tuner distributes file requests amongst every peer that responds during the Channel Peer phase.

If you set the value of this property to a negative value or a non-integer value, the tuner tries to get the files from all the peers which are ready to send files.

If you set this property to 0, the tuner gets files from the Transmitter irrespective of the number of peers ready to send files during the Channel Peer or File Peer phase. If you set this property to 1, the tuner gets files only from a single peer, irrespective of the number of peers ready to send files during the Channel Peer or File Peer phase.

If you set this property to any value more than 1, then the tuner obtain files only from the specified number of peers. For example, if you set the value of this property to 4, and 20 peers are ready to serve during the Channel Peer phase, the tuner obtains files only from 4 peers. The tuner does not contact the remaining 16 peers that are also ready to serve files.

This property does not affect the File Peer phase of MESH. The default value of this property when not set is 5.

`marimba.tuner.p2p.no.retries`

specifies the number of retries which a tuner attempts to request for a file.

This property is deprecated.

`marimba.tuner.scheduler.checkonlinestatus`

specifies that when you set this tuner property to true, the tuner delays scheduled updates for channels until the network is available. You can use this property in scenarios where the network is not immediately available after a computer has exited from the sleep mode. This tuner property prevents channels that have no access to the transmitter from being updated on schedule.

`marimba.tuner.deprecate64bitsegment`

While implementing the 64-bit tuner, you can toggle between a 64-bit tuner and a 32-bit tuner using this property.

Following are the possible values:

- If the value is true, the tuner brings x86(32-bit) segments from the transmitter.
- If the value is false, the tuner brings x64(64-bit) segments from the transmitter.

`marimba.tuner.jre.arch`

specifies the architecture of the tuner jre.

Following are the possible values:

- 32

■ 64

Note: This property is not configurable.

`marimba.tuner.process.priority`

specifies the priority you must set for java.exe and minituner.exe processes.

Following are the possible values:

- high
- normal
- idle

`marimba.tuner.interactiveservicesdialog.enable`

By default, in Windows Server 2012 and Windows 8, Interactive Service Detection Service is disabled. If you want to access a fully interactive tuner running in service session, you must enable this service. It allows the Interactive Service dialog box to switch to service session.

This property specifies whether you need to enable the Interactive Service dialog box in user session.

- Valid values: True or False
- Default value: False

Note: After you change the values of the properties, restart the tuner.

For more information on usage of this property, refer to “Advanced tuner settings” section in Chapter 20 of Symphony Marimba Client Automation CMS and Tuner User Guide.

`marimba.tuner.administrator.port`

specifies the port on which the tuner needs to be administered using “Lite Weight Tuner Administrator” feature. Default value: 7799

Note: After you change the values of the properties, restart the tuner.

`marimba.reboot.interact.snooze.maxLimit`

specifies the number of times a user is allowed to defer the reboot after the configured time is elapsed. The default value of this property is 0.

`marimba.tuner.jce.unlimitedstrength`

With Kerberos authentication, we use Java JCE (Java Cryptography Extension) for encrypted message exchange.

The tuner (out of the box) contains the following JARs:

- local_policy_original.jar [from JRE/lib/Security folder]
- US_export_policy_original.jar [from JRE/lib/Security folder]
- Local_policy_new.jar [downloaded from the oracle site]
- US_export_policy_new.jar [downloaded from the oracle site]

The tuner will NOT contain the "local_policy.jar" and "US_export_policy.jar" out of the box JAR files. The tuner launcher will add these files based on this property.

If property is set to "false" or not set, the launcher will copy the original JAR files into local_policy.jar and US_export_policy.jar

If property is set to "true": the launcher will:

- Infrastructure Service will "ignore" the security related JARs during Infrastructure Service upgrade cycle
- Integrity check code will similarly "ignore" the security related JARs during Infrastructure Service upgrade cycle

`marimba.tuner.startmenu`

specifies whether the user wants to start the primary channel from the start menu. This will help user to always start the primary channel in the first click from the start menu even when tuner wakes up from the minimal mode.

`runtime.ssl.provider`

indicates the SSL implementation used by the tuner.

Valid value: rsa, java, native, custom

`runtime.tuner.intelamt`

specifies the state of the Intel Active Management Technology (Intel AMT) chip on the machine.

Valid value: notpresent (The chip is not available on the machine), presentdisable (The chip is available on the machine, but not enabled for use), presentenable (The chip is available on the machine, and the chip is enabled for flash storage)

`runtime.verbose`

determines the amount of logging information produced by the tuner. The property is by default set to `false`, which is the recommended setting. Setting the property to `true` can result in a large volume of logging data sent to the database. Setting this property to `true` is required, however, if you want to collect and query for log messages with the severity level of `INFO`.

Runtime properties, such as `runtime.verbose`, are set by the tuner, and users should not set them directly in the `prefs.txt` file. To set this property, use one of the following approaches:

- For Windows, start the tuner with the `-v` command-line option, and set the tuner property called `marimba.launch.args` to `-v` (which indicates verbose mode). If the tuner is running as a service, you can set the `marimba.launch.NTServiceArgs` to `-v` also.
- For UNIX, start the tuner with the `-v` command-line option.

Valid value: `true` or `false`

Default value: `false`

`snmp.manager.port`

specifies the Infrastructure Status Monitor SNMP manager port or 3rd party SNMP manager port. You can change the value of this property through profiles, Policy Management, or Tuner Administration. This property is not valid on repeaters.

Valid value: integer

Default value: 162

You must restart the Logging plug-in for the new value of this property to take effect. You can restart the Logging plug-in in the following ways:

- When using a profile to change value of this property, the tuner is restarted automatically and the new settings are applied to the plug-in during the next update schedule of the Infrastructure Service. This is the recommended method for changing the SNMP manager port.
- Re-publish the Logging plug-in through Report Center.
- Restart the tuner on which the transmitter is configured to host the SNMP manager.

`trigger.update.wakeup`

specifies that when you set this channel property to `any` irrespective of how the computer was woken up, after a computer wakes up and the network is available, the tuner automatically starts the channel and updates.

If you set the value of this channel property to `wow`, after a computer wakes up and the network is available, and if the computer performed an unattended wakeup (wakeup through Wake On Lan (WOL)), then the tuner automatically starts the channel and updates.

`marimba.tuner.amt.user`

Contains the AMT username of the vPro machine

`marimba.tuner.amt.password`

Contains the AMT password of the vPro machine. You can use these username and password to access the vPro settings of an AMT enabled machine. If you set the password through Tuner Admin custom properties, commandline, or Tuner prefs file, it must be set in Base64 encoded format.

`marimba.tuner.amt.localaccess`

To enable local vPro API access (e.g. Agent Presence, Storage), set this property to `true`.

If the property is set to `true` and you need access to amt storage, a watchdog must be created.

If you set this property to `false` or you do not set this property, local vPro API access is disabled. However, the remote APIs (like vPro Wakeup, PC Alarm Clock, etc.) continue to work.

`loggerplugin.inserter.url`

On the Mirror or Master transmitter, set this tuner property to configure the forwarding of log reports, SNMP alerts, stats to the Inserter.

The value of this property should be the URL of Logging Service channel published on Inserter Transmitter.

marimba.tuner.channels.dcap

- This property maintains a list of channels that need to be blacklisted. Any attempt made to start this channel will fail. The channels specified in the property must be in Base 64 encoded format and can be separated using semi-colon(;) .

e.g.

marimba.tuner.channels.dcap=Q2hhbm5lbE1hbmFnZXI=;Q2VydGlmaWNhdGVNYW5hZ2Vy

That means we are putting ChannelManager and CertificateManager in the blacklist.

Note: This is case sensitive. And if a channel is once blacklisted, it will be blacklisted till tuner gets restarted (even after changing the property). This feature is applicable to all OSes.

marimba.tuner.native.delaystart

This property delays the start of tuner launcher.

If the value of this property is greater than 0, launcher would wait for those many seconds at the launch time.

marimba.tuner.java.delaystart

This property delays the start of tuner.

If the value of this property is greater than 0, Tuner would wait for those many seconds at the launch time.

Note: Feature for launcher (tuner.exe) is applicable only for windows (And that is intentionally designed). The feature for java.exe (main tuner) is applicable for all OSes. It is not applicable for mini tuner.

`marimba.reboot.perday.limit`

This property restricts the number of times a machine can be rebooted in a day.

It takes integer value equal to or greater than 0. If the value of this property is set to 0, the machine cannot be rebooted for the day. If the value of this property is set to 1, then the machine can be rebooted only once in a day.

Note: This property restricts users from rebooting the machine through Marimba application. It doesn't restrict users to reboot the machine manually. However, CRS keeps track of such manual reboots and the per day count is handled accordingly.

Chapter

3

Proxy properties

You have the option of using proxy properties to configure a proxy if you cannot do so using profiles, the Proxy Administrator graphical user interface, or the command-line interface.

The following topics are provided:

- Overview of proxy properties (page 228)
- List of properties (page 229)

Overview of proxy properties

Normally, you do not need to configure a proxy using proxy properties. You do so only if you cannot configure a feature using profiles, the Proxy Administrator graphical user interface, or the command-line interface. If the need arises to add proxy properties to your configuration, follow the guidelines in the documentation or those from Professional Services or Marimba Channel Store.

You can also set additional generic channel properties for the proxy. For a list of these properties, see “Channel properties” on page 247.

You can set proxy properties either using profiles or using Proxy Administrator browser-based interface. *In general, you should set proxy properties using profiles.* This is because properties set using Proxy Administrator always override those set using profiles. Thus, if you set properties A and B using profiles, and you then change the values for these two properties using the Proxy Administrator, the latter values always take effect. Even if you update the profile or apply a new profile to the machine *after* you made the changes using Proxy Administrator, the values set by the Proxy Administrator always take precedence over the values specified in a profile.

For more information about when to use profiles versus Proxy Administrator, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.

For instructions about setting proxy properties, refer to the following documentation:

- To set proxy properties using profiles, see the online help for the Profiles tab in Setup & Deployment. To access the help, from the Applications menu on your CMS Console, select Infrastructure > Setup & Deployment, click the Profiles tab, and then click Help. Perform a search for proxy properties.
- To set proxy properties using Proxy Administrator, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.
- To set proxy properties using the Proxy Administrator Command Line channel, see “Proxy Administrator options” on page 103.

Proxy properties are stored in different places, depending on how you set them. If you set proxy properties using Proxy Administrator, they are saved in the `properties.txt` file in the proxy's workspace. If you set proxy properties using profiles, they are saved in the `application.txt` file in the proxy's channel directory (located inside the tuner's workspace). Proxy properties set using Proxy Administrator always override those set using profiles. If the situation arises where a property has been set in the `properties.txt` file, and you need the profile property in the `application.txt` to take effect, contact your Professional Services or Customer Support representative.

Note: Proxy channel properties are different from tuner proxy properties.

The proxy channel properties configure your proxies. On the other hand, the `marimba.proxy.<option>` tuner properties configure clients (such as endpoint tuners) to use a normal proxy. Do *not* configure these tuner properties for the proxy's tuner. For information about proxy tuner properties, see “Tuner properties” on page 165.

List of properties

The following sections list the proxy channel properties.

Main proxy properties

This section lists the main proxy properties.

`proxy.adminPermissions`

specifies the base-64 encoded user name and password that must be used to administer the proxy.

To specify the administration user name and password, use profiles, the Proxy Administrator GUI, or the Proxy Administrator command-line interface. For more information, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.

`proxy.enableTaskBarIcons`

(Windows only)

specifies whether the taskbar icon for the proxy () is displayed when the proxy channel is running on Windows machines.

Valid value: true or false.

`proxy.inheritTunerPassword`

specifies whether the proxy should inherit the administration credentials (that is, the user name and password) of the tuner on which the proxy is running.

Note: You cannot inherit the tuner's administration credentials if these credentials are authenticated using LDAP.

Default value: `false` (the tuner's administration user name and password are not inherited)

`proxy.outgoingHost`

specifies the outgoing host name or IP address to use for network connections to other machines. This property is required if the proxy is installed on a multi-homed system.

`proxy.root`

specifies the location of the proxy's root directory.

Server proxy properties

This section lists the server proxy properties.

`server.connect.active`

specifies the number of simultaneous connections allowed to the proxy.

Default value: 1024

`server.connect.backlog`

specifies the number of connections that are waiting to be serviced by the proxy.

Default value: 50

`server.connect.ipaddr`

specifies the bind address for the proxy's listener port. This property is required if the proxy is installed on a multi-homed system.

`server.connect.port`

specifies the proxy's listener port number.

Default value: 8080

server.connect.rate

specifies how much network bandwidth is used to service proxy requests (for example, for obtaining or caching content from a transmitter). As part of its request, the proxy informs the transmitter the bandwidth it should use to service the proxy's request.

Default value: 0 (the maximum available bandwidth is used)

server.proxychain

specifies whether the proxy is chained to another proxy. The next proxy in the chain is specified by the `server.proxychain.nextProxy` property. You can include only normal proxies in a proxy chain.

server.proxychain.nextProxy

specifies the next proxy in a proxy chain in the form `<host>:<port>`. Also, to enable proxy chaining, you must also set the `server.proxychain` property to true. You can include only normal proxies in a proxy chain.

server.reverse

specifies whether the proxy should run as a reverse proxy. Specify the target transmitter (the internal transmitter) for the reverse proxy using the `server.reverse.target` property.

server.reverse.secure

specifies whether the proxy should run as a secure reverse proxy. Also specify the certificate ID of the SSL certificate you are using and the password for the certificate.

server.reverse.secure.certid

specifies the unique certificate ID of the SSL certificate to use for the proxy. To find this ID string, start Certificate Manager, select the certificate, and click View.

server.reverse.secure.certpw

specifies the private key password for the certificate specified in the `server.reverse.secure.certid` property. This property is set only if the proxy is started with the `-savesslpw` option.

Valid value: a base64-encoded string

server.reverse.secure.clientAuth

specifies the client authorization mode for the reverse SSL listener.

Valid value: none, request, and require

Default value: none

`server.reverse.secure.savepw`

specifies whether the password for the SSL certificate is saved on the proxy's machine for autostart capabilities.

Use the Proxy Administrator browser-based interface to configure this feature.

Valid value: true or false.

`server.reverse.secure.strongEncryption`

specifies whether you want to use strong (or domestic, 128-bit) encryption strength for the secure reverse proxy. You should not need to use this property.

Valid value: 128-bit encryption strength

`server.reverse.target`

specifies the URL of the target transmitter (the internal transmitter) for the reverse proxy. To configure a proxy as a reverse proxy, also set the `server.reverse` property to true.

`server.transfer.buflen`

specifies the transfer buffer size in bytes.

Default value: 8192

Cache proxy properties

This section lists the cache proxy properties.

`cache.lowWaterMark`

specifies a percentage, representing the low watermark for cache garbage collection (which stops when the cache reaches this size). Set the cache low watermark to be about 75% - 80% of the maximum cache size.

Default value: 75

`cache.maxSize`

specifies the maximum cache size for the proxy in MB. Specify a maximum cache size that is (at the most) 90% of the disk space. Remember that the cache garbage collection process starts if the total amount of disk space on the proxy's machine is less than 10%.

Default value: 1024

Refresh proxy properties

This section lists the refresh proxy properties.

`refresh.credentials`

specifies the subscribe credentials (that is, the user name and password) for one or more channels. This property is required if you want to preload channels from a transmitter that has subscribe permissions configured.

Specify the value for this property in plain text using the following format:
`plain:<user>:<password>`

Important: You *must* include `plain:` in front of the user name and password.

Also, you must provide *both* a user name and a password.

Although you specify the subscribe user name and password in plain text on the command-line, they are stored as a base64-encoded string in the proxy's `properties.txt` file.

The *same* credentials specified using the `refresh.credentials` property are used for *all* the channels that you are preloading at one time. If different channels require different user names and passwords, preload the channels in multiple phases. In this case, specify the correct user name and password each time before preloading the content.

Use the Proxy Administrator browser-based interface to set this value. Alternatively, if you want to set this value using the Proxy Administrator Command Line channel, see the `-setproperty` option using the following format:

```
runchannel <ProxyAdmin_channel_URL>
[<user_name>:<password>@]<host>:<port> -setProperty
refresh.credentials plain:<user>:<password>
```

Log proxy properties

This section lists the log proxy properties.

`log.directory`

specifies the location of the log directory. If the directory does not already exist, it is created automatically.

```
log.access.roll.policy  
log.admin.roll.policy
```

specifies how often the proxy closes the current log file and creates a new one.

The main reason to roll a log is that one huge file, in contrast to many small files, is hard to manage. Another reason for rolling logs is that as long as an application is running and writing to its logs file, an external program cannot access that file. You have to stop the application to access the log file.

Specify one of the following roll policies:

- By date, which can be one of the following:

- hourly
- daily
- weekly
- monthly
- yearly

- `bysize` — The log file rolls as soon as it reaches a certain size.
- `never` — The log file never rolls; the current log file never closes. So, there is one log file.

Default value: `weekly`

```
log.access.roll.size  
log.admin.roll.size
```

specifies a size in KB when the proxy starts a new log file. When this maximum size is reached, the current log file is closed and a new one is created.

You can specify this property only if you select the `bysize` policy.

Default value: `512`

```
log.access.roll.versions  
log.admin.roll.versions
```

specifies the number of previous rolled log file versions (in addition to the current one) that the proxy keeps.

When the maximum number is reached, the oldest log file is deleted automatically. For example, if you have the maximum file count set to 3, then when the 4th log file is created, the oldest is deleted.

Note: If you specify never as the log rolling policy (using the `log.access.roll.policy` or the `log.admin.roll.policy` property), the log roll versions do not take effect.

Default value: 4

Chapter
4

Transmitter properties

You have the option of using transmitter properties to configure a transmitter if you cannot do so using profiles, the Transmitter Administrator graphical user interface, or the command-line interface.

The following topics are provided:

- Overview of transmitter properties (page 238)
- List of properties (page 239)
- Transmitter extension properties (page 244)

Overview of transmitter properties

Normally, you do not need to configure a transmitter using transmitter properties. You do so only if you cannot configure a feature using profiles, the Transmitter Administrator graphical user interface, or the command-line interface. If the need arises to add transmitter properties to your configuration, you are advised to do so in the documentation or by Professional Services or Customer Support.

You can also set additional generic channel properties for the transmitter. For a list of these properties, see “Channel properties” on page 247.

You can set transmitter properties either using profiles or using Transmitter Administrator browser-based interface. *In general, you should set transmitter properties using profiles.* This is because properties set using Transmitter Administrator always override those set using profiles. Thus, if you set properties A and B using profiles, and you then change the values for these two properties using the Transmitter Administrator, the latter values always take effect. Even if you update the profile or apply a new profile to the machine *after* you made the changes using Transmitter Administrator, the values set by the Transmitter Administrator always take precedence over the values specified in a profile.

For more information about when to use profiles versus Transmitter Administrator, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.

For instructions about setting transmitter properties, refer to the following documentation:

- To set transmitter properties using profiles, see the online help for the Profiles tab in Setup & Deployment. To access this online help, from the Applications menu on your CMS Console, select Infrastructure > Setup & Deployment, click the Profiles tab, and then click Help. Perform a search for transmitter properties.
- To set transmitter properties using Transmitter Administrator, see the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.

Both profiles and Transmitter Administrator do not offer you a way to delete transmitter properties after you add them. Therefore, use caution when adding properties.

Transmitter properties are stored in different places, depending on how you set them. If you set transmitter properties using Transmitter Administrator, they are saved in the `properties.txt` file in the transmitter's workspace. If you set transmitter properties using profiles, they are saved in the `application.txt` file in the transmitter's channel directory (located inside the tuner's workspace). Transmitter properties set using Transmitter Administrator always override those set using profiles. If the situation arises where a property has been set in the `properties.txt` file, and you need the profile property in the `application.txt` to take effect, contact your Professional Services or Customer Support representative.

List of properties

This section lists the transmitter properties.

`main.clientCertPassword`

specifies the certificate password for a mirror or repeater. If the master transmitter requires a client-side certificate from mirrors or repeaters, install a certificate on the mirror or repeater, and set the certificate password using this property.

`main.transmitter.root`

specifies the root directory (or workspace location) for the transmitter. Changing the transmitter's root directory (also called the transmitter's workspace directory) does not copy or move the contents of your current transmitter root directory. It only points the transmitter to a new root directory. You must copy the contents of your current transmitter root directory. However, any new changes are recorded only in the new root directory.

`main.segments.url`

specifies the URL or the file path of the custom segments XML file.

`repeater.schedule.restricted`

specifies whether the times when a mirror or repeater synchronizes with its master transmitter (or source transmitter) are restricted. By setting this property to `true` on the mirror or repeater, the mirror or repeater replicates content from its master except in the time period specified in the `repeater.sync.blackout` property.

`repeater.sync.blackout`

specifies a blackout window during which synchronization between the transmitter and source transmitter is blocked, regardless of the replication interval. If the `repeater.schedule.restricted` property is set to “`false`,” then the schedule specified in the `repeater.sync.blackout` property is ignored, and replication continues as specified in the replication interval. Any change to the property restarts the Repeater Module only.

Examples:

`repeater.sync.blackout=never`: Specifies no blackout window.
Replication continues with its replication interval.

`repeater.sync.blackout=anytime on mon+tue`: Specifies a blackout period all day on Monday and Tuesday when replication is blocked.

`repeater.sync.blackout=between 09:00AM and 05:00PM on mon+tue+wed`: Specifies a blackout period between 9 AM and 5 PM on Monday, Tuesday, and Wednesday when replication is blocked.

`stats.days`

specifies the number of days of status information that is kept. Set this property for all status reports. For more information, see `stats.enabled`.

Valid value: an integer (time in days)

Default value: 7

`stats.enabled`

returns a report containing status and statistics about a transmitter, either a one-time request or automatically on a schedule. Transmitter status can contain information about the transmitter, tuner, virtual machine, operating system, and storage. Statistics can include the number and type of requests, the average time per request, the bytes sent and received, and errors, all aggregated by hour.

Also set the `stats.days=<number of days>` property to return a status report. Set the `stats.report.*` properties to have the transmitter post the status reports on schedule.

Valid value: true or false

Default value: true

`stats.report.enabled`

specifies whether the transmitter posts status reports on schedule. For more information, see `stats.enabled`.

Valid value: true or false

Default value: false

`stats.report.http.url`

specifies information relating to the transmitter posting schedule. Set this property to have the transmitter post the status reports on schedule. For more information, see `stats.enabled`.

Valid value: a URL

Default value: `http://<Web server URL>` where `<Web server URL>` is a Web server that can receive and process posts on the URL. This may require a custom script on the Web server.

`stats.report.interval`

specifies information relating to the transmitter posting schedule. Set this property to have the transmitter post the status reports on schedule. For more information, see `stats.enabled`.

Valid value: an integer (time in seconds)

Default value: 1800, or receiving a report every 30 minutes

`stats.report.type=http`

specifies information relating to the transmitter posting schedule. Set this property to have the transmitter post the status reports on schedule. For more information, see `stats.enabled`.

`transmitter.http.bindAddress`

specifies the interface that the transmitter listens on for HTTP connections (and the interface that you must specify to get a listing, subscribed channels, channel updates, and so on). If the property is not set, then the transmitter (Java VM) binds to all interfaces. If you are running a transmitter on a multi-homed machine—with multiple network interface cards (NICs), you can specify the interface that the transmitter should bind to for incoming and outgoing requests.

`transmitter.http.externalURL`

configures the transmitter to send the specified IP address or host name to clients, instead of its own. Set this property for mirrors or the master transmitter, for example, when you have set up a load balancing scheme in your environment, and you need the master or mirror to return the IP address or host name of the load balancer to clients. In this case, on each transmitter (master and mirrors) behind the load balancer, set this property to the IP address or host name of the load balancer.

Configure the master and mirrors in this way when you have a redirection strategy configured where you want the master and mirrors to be included in the list of repeaters provided to clients. In such a setup, if the master or mirror returns its own information to the client, the client tries to bypass the load balancer and contact that transmitter directly (sometimes resulting in a failed attempt). By setting this property for the master and mirrors located behind the load balancer, the clients contact the load balancer (and then are directed to one of the transmitters).

`transmitter.http.outgoing.host`

specifies the interface that outbound HTTP requests use. If you are running a transmitter on a multi-homed machine—with multiple network interface cards (NICs), you can specify the interface that the transmitter should bind to for incoming and outgoing requests. This property is often necessary if you have a multiple repeaters or mirrors installed on the same machine. This way, requests from a repeater or mirror consistently come from a single interface, and the master knows the IP address to redirect clients to.

`transmitter.http.timeout`

represents the timeout period in seconds for a transmitter HTTP connection. The default timeout is 60 seconds.

Valid value: an integer (time in seconds)

Default value: 60

`transmitter.logs.useclientip`

specifies whether the endpoint's IP address or the load balancer's (or proxy's) IP address is printed to the access log of the transmitter.

If true then the IP address of the endpoint is printed in the access logs of the transmitter, even if the endpoint is behind a proxy or load balancer.

If false then IP address of the nearest accessing node (i.e. either proxy or load balancer) is printed in the access logs of the transmitter.

Default value: false

`transmitter.ptree.verify.enabled`

performs the task of segment verification when the segment is published. By default, this property is set to false which means that the segment or ptree verification task is not performed when the channel is published.

Valid value: true or false

Default value: false

`transmitter.segmentation.hierarchy`

specifies the type of segmentation hierarchy which the transmitter follows. You can also edit the segmentation hierarchy that the transmitter follows in the Transmitter Administration page.

Note: The segmentation hierarchy is applicable to all the channels and packages published on the Master Transmitter. On a Mirror Transmitter, this hierarchy is replicated from the source from where it replicates the contents and is applicable to all the channels and packages published on the source. On a Repeater, this hierarchy is applied to all the channels and packages published on the Repeater. The segmentation hierarchy value that it replicates from its source is stored internally.

Valid values: default, flat, and custom

Default value: default

Transmitter extension properties

This section lists properties for the extensions of the transmitter—such as the Subnet Repeater Policy extension.

`allow.arbitrary`

specifies whether client tuners are redirected to the list of repeaters specified in the `config.xml` file. However, if the client cannot contact the repeaters in the list, the mirror or master handles the request unless it has been configured not to do so. In this case, the mirror or master returns an error message.

You can force the subnet repeater policy to deliver a list of repeaters to the tuner even if the transmitter has no knowledge of those repeaters. (This is the default behavior for the subnet repeater policy, and overrides the default transmitter redirection behavior. For more information about how redirection works, see the appendix of the *Symphony Marimba Client Automation Transmitter and Proxy User Guide*, available on the Marimba Channel Store.

Forcing a transmitter to return a set list of repeaters is useful when several mirrors, and possibly the master transmitter, are being used in a load-balancing scheme. For example, several mirrors, and possibly the master, are located behind a load balancer, and repeaters are connecting to the load balancer. A master or mirror can be certain that all repeaters are functioning only if each repeater makes regular contact for updates. In the case where mirrors, and perhaps the master, are used in a load-balancing scheme, it is likely for at least one of these transmitters to assume erroneously that one of the repeaters has failed simply because it has been contacting one of the other transmitters. By allowing a master or mirror to return a certain list of repeaters even when it assumes that one of those repeaters has failed, all repeaters can be used to full advantage. Endpoint tuners still contact the master or mirror to refresh their repeater list after a certain time interval, but the master or mirror always returns the list of repeaters that is specified by the subnet repeater policy.

To enforce the subnet repeater policy, enable the allow arbitrary mode by setting the following key-value pair in the `parameters.txt` file (which is included in the `Subnet_Repeater_Policy.zip` file):

`allow.arbitrary=true`

Valid value: true or false

Default value: true

natok

specifies whether the subnet repeater policy retrieves the client's IP address from the profile information in the client's request or from the incoming socket.

You can specify the IP address the subnet repeater policy should use to determine the repeater list for a client. This parameter is useful, for example, when a client is going through a proxy to communicate with the transmitter. In this case, you can use this parameter to specify whether the subnet repeater policy should use the IP address of the client or of the proxy.

If you set this property to `false`, the subnet repeater policy retrieves the client's IP address from the profile information in the client's request. However, if profiles are not enabled on the client, the subnet repeater policy uses the actual IP address of the incoming socket. If the client is going through a proxy, the subnet repeater policy uses the IP address of the proxy.

If you set this property to `true`, the subnet repeater policy always uses the IP address of the incoming socket. If the client is going through a proxy, the subnet repeater policy uses the IP address of the proxy. Thus, set this property to `true` if, for example, a client is using a proxy, and profiles are enabled on the client, but you want the subnet repeater policy to use the IP address of the proxy, not the IP address of the client.

Valid value: true or false

Default value: false

Chapter

5

Channel properties

Channel properties qualify the appearance and behavior of channels.

The following topics are provided:

- Overview of channel properties (page 248)
- List of properties (page 248)
- Properties for Application Packager channels (page 262)
- Properties for the Policy Manager channel (page 266)
- Properties for the Tuner Update Manager channel (page 266)

Overview of channel properties

When you assign properties to a channel, Channel Copier (and Publisher) creates a `properties.txt` file in the channel directory. This file is published with the channel and is used to communicate settings to the tuner kernel. It specifies the main class for the channel, the update schedule, and so on. You can edit this file, although Channel Copier, Publisher, and Application Packager can overwrite your settings.

Entries in the `properties.txt` file have this form:

`<channel_property>=<property_value>`

For example:

`mimetype=application/x-castanet-channel`

Note: You can change some channel properties through the graphical user interface (GUI), while Marimba products set others, and changing them manually is not recommended.

List of properties

This section lists the some of the properties that you can assign to a channel with Channel Copier (and Publisher).

`active.name`

is the nameservice name for this instance of the channel. This property is set when the channel is started, and the transmitter uses this property to find its console.

Valid value: a string

`admin`

is the administrator name.

Valid value: a string

`author`

is the author of the channel.

Valid value: a string

`browser.url`

is the relative URL for an HTML channel to act as a browser. If this property is not set, the tuner browser is used.

Valid value: a URL

`capabilities`

sets the privileges for the channel.

Valid value: none or all

`category`

is the category into which the channel is grouped. The tuner lists the title of the channel in the specified category. The only well-defined value is hidden, which indicates that the channel shouldn't be shown in the channel list. Channel Manager uses this property.

Valid value: a string

`channel.version`

is the version of the channel.

Valid value: a string

`channel.p2p.enabled`

determines whether the channel is updated using MESH or not. When you set the value of this property to true or if this property is not set, the channel updates using the MESH technique which is configured using the marimba.tuner.p2p.enabled MESH tuner property. If you set the value of this property to false, the channel does not update using MESH technique.

This property does not prevent the ability of a MESH-enabled tuner to work as a serving peer, and hence this property is applicable only when a MESH-enabled tuner requests for a channel or a package.

This property is applicable only for channel updates. Channel subscribers are dependent on the configuration of the marimba.tuner.p2p.enabled property.

Valid value: true or false

`classpath`

is the path to the Java class files. This property is used when loading the channel.

Valid value: the path name

code

is, for applet channels, the main class for an applet. AppletPanel uses this property.

Valid value: the class name

codebase

is, for applet channels, the code base for an applet. This property is part of the classpath.

Valid value: the applet code base

copyright

is the copyright for the channel.

Valid value: a string

delete.alert

indicates whether the user is alerted when the channel is being deleted.

Valid value: true or false

description

is the description of the channel.

Valid value: a string

extension

is the file extension to use in the transmitter listing for Internet Explorer.

Valid value: a string

filepackager.savemanifest

enables, when set to true, File Packager to save any changes you make (for example, changing the installation policy) when editing a File Packaged channel and to apply those changes when you repackage the channel.

When set to false, or if this property is omitted from the properties.txt file, changes you make when editing a File Packaged channel may not be saved when you repackage the channel. This property applies to Application Packager version 4.6.1.5 and later.

Note: When repackaging, instead of using this property, you can use the XML template files described in the *Application Packager User Guide*, available on the Marimba Channel Store.

Valid value: true or false

Default value: false

`filepackager.setnewtimestamp`

captures, when set to true, the new time stamp attributes of files modified during repackaging. The `filepackager.savemanifest` property must also be set to true.

Valid value: true or false

Default value: false

`filepackager.setotherattributes`

captures, when set to true, the new attributes (except for time stamp) of files modified during repackaging. The `filepackager.savemanifest` property must also be set to true.

Valid value: true or false

Default value: false

`frame.bounds`

is the location and size of the channel window. LaunchFrame uses this property.

Valid value: integer x, integer y, integer width, integer height

`height`

is the height of the window in pixels for applet channels. AppletPanel uses this property.

Valid value: an integer

`hide`

indicates whether the channel title is hidden from the transmitter listing in the tuner.

Valid value: true or false

icon

is the path to the channel's window icon. LaunchFrame uses this property. A 50x50 pixel .gif file icon used on UNIX.

Valid value: the path name

icon.smaller

is the path to the channel's window icon. LaunchFrame uses this property. A 32x32 pixel icon used on Windows.

Valid value: the path name

index.page

is the relative URL of the starting page of an HTML channel.

Valid value: a URL

install.active

is used only if update.action is not set. This property has the same format as update.action.

Valid value: ignore (tuner does nothing), restart (restarts the channel), or install (automatically installs new data).

install.inactive

is the action the tuner kernel takes when new data is available, but the channel is not running.

Valid value: ignore (tuner does nothing), start (starts the channel if new data arrived), notify (sends a notification to the channel if new data arrived, but does not start the channel), or notifyalways (always sends a notification to the channel, regardless of new data arrival).

interrupt

indicates whether the tuner waits for this channel to stop before restarting the tuner.

Valid value: true (the tuner does not wait for the channel to stop before restarting) or false (the tuner waits for the channel to stop before restarting. The maximum amount of time that the tuner waits is specified using the tuner property marimba.tuner.restart.timeout).

Default value: true

iphost.expire

indicates when the IP address associated with a channel expires. This address is the IP address of the source transmitter from which a channel was downloaded. The expire policy is checked, and accordingly, the IP address can refresh the next time this channel updates. If not set, the tuner property `marimba.tuner.iphost.expire` is the default.

Valid value: -1 (the IP address never expires. The DNS lookup is repeated only if the old IP address breaks), 0 (the IP address for the channel is not cached. Thus, the IP address is expired. Each time the channel is updated, the DNS lookup is repeated.), or <min> (the IP address expires after the specified number of minutes. The next time the channel is updated, the DNS lookup is repeated only if the current IP address expired according to the <min> setting.)

Default value: -1

locale

is the channel locale.

Valid value: see Table 5-2 on page 261.

logon.action**logoff.action**

is the action the channel performs when a user logs on or logs off and the tuner receives notification. You must set `logon.notify` or `logoff.notify` to true for the channel.

Valid value: update, start, remove, unsubscribe, or updatestart (update and then start the channel)

Default value: start

logon.args, logoff.args

is the argument the channel starts with when a user logs on or logs off and the tuner receives notification. You must set `logon.notify` or `logoff.notify` to true for the channel.

Valid value: the argument the channel starts with (for example, `-install`)

logon.notify, logoff.notify

is, when set to true, the action the tuner performs (specified using `logon.action` or `logoff.action`) when a user logs on or logs off and the tuner receives notification.

Valid value: true or false

logs.arguments

prevents, when set to `false`, the arguments (such as passwords provided with command-line options) from appearing in the log files.

Valid value: `true` or `false`

main

is the name of the main class for the channel. The Publisher usually sets this property, but the tuner can make substitutions for well known classes such as `ApplicationPlayerFrame`.

Valid value: a class name

mimetype

is the MIME-type to return for channels in the transmitter listing.

Valid value: the MIME type

name

is the name of the channel to display on the tuner. Deprecated. This property is replaced with `title` and ignored if that property is set.

Valid value: a string

network.notify

starts the channel if the following conditions are true:

- the value is set to `true`
- the scheduler is active and the channel is not in a blackout period nor exempt from one,
- the tuner detects that the network is available, including if it had to wake up from minimal mode

When the `network.notify` channel property is set to its default value of `true`, and when the tuner goes from offline to online, the Infrastructure Service starts, even if the update schedule is set to `Never`.

Valid value: `true` or `false`

Default value: `true`

notifyremove

specifies, when set to true, that the channel sends an APP_REMOVE notification when it is deleted from the tuner, causing the application associated with the channel (for example, a channel packaged with Application Packager) to be uninstalled from the machine. When set to false, the channel does not send an APP_REMOVE notification when it is deleted from the tuner, causing the application associated with the channel to remain on the machine even when the channel is uninstalled.

Valid value: true or false

p2p.file.nopeers

Tracks the number of peers that delivered files to the requesting tuner. This property is set in the requesting peer tuner's channel.txt file located in the folder of the channel which is subscribed through Mesh.

p2p.nopeers

Determines the number of peers that the requesting peer visited to get files. However, the requesting peer tuner may not get files from all these peers. This property is set in the requesting peer tuner's channel.txt file located in the folder of the channel which is subscribed through Mesh.

pickup.tunerprops

Specifies whether the Inventory Service running on the device scans the tuner properties from prefs.txt and properties.txt.

Valid value: true or false

platform

is the platform for the channel.

Valid value: see Table 5-1 on page 260.

presentation

is the presentation file name for a Bongo presentation channel. ApplicationPlayerPanel uses this property.

Valid value: a short file name

publish.time

is the time (in milliseconds since epoch) when the channel was published.

Valid value: a long integer

scanner.services.rcstartmsgCMD.enable

After adding this property to the application.txt file, it enables the Inventory Service to capture all of the system service attributes on HP-UX. Enabling this property can cause some applications to restart while scanning the HP-UX machine.

Default value: false

security.sslOnly

specifies, when set to true, that the channel is served only when the tuner connects over SSL.

Valid value: true or false

service.autostart.order

is the autostart order for the channel that indicates this channel is a service that should start at tuner startup. Lower values start earlier.

Valid value: an integer

service.daemon

specifies, when set to true, that the channel runs as a daemon. If the tuner detects that all currently running channels are daemon channels, it behaves as if no channels are running.

Valid value: true or false

service.ondemand.name

specifies that a channel can act as a particular service. The kernel starts this channel (if necessary) when a user asks for the service.

Valid value: an on-demand service name for the channel

signing.certkey

is the unique identifier for the signing certificate.

Valid value: a Cert key ID

signing.enabled

indicates whether the channel is signed.

Valid value: true or false

signing.scope

indicates whether the entire channel is signed or just the *signed* directory.

Valid value: all or signed

start.schedule

is the schedule at which the channel should start.

Valid value: see “Syntax for the schedule string” on page 259.

start.schedule.skipfirst

specifies, if set to `false`, that channels with a start schedule that is cyclic (for example, every 2 days update at 4:00am) are started immediately after being subscribed. These channels should complete important, periodic work when the channels are subscribed. When you do not want this behavior, set this property to `true` to suppress the first execution of the channel.

Valid value: `true` or `false`

Default value: `false`

subscription.schedule

uses Policy Manager to set the Scanner Service and Logging Service channels `update.schedule` property and `subscription.schedule` property to `true`, and to spread out the inventory update schedule so that all your endpoints do not send scan reports at the same time and put a heavy burden on the transmitter.

Valid value: `true` or `false`

thumbnail

is the path to a 16x16 .gif thumbnail image for inactive channels.

Channel Manager, and sometimes LaunchFrame, use this property.

Valid value: a path name

thumbnail.running

is the path to a 16x16 .gif thumbnail image for a channel that is running.

Channel Manager uses this property

Valid value: a path name

title

is the display title of a channel. Channel Manager uses this property.

Valid value: a string

type

is the kind of information contained within the channel. This property is used to determine how to start the channel.

Valid value: applet, application, presentation, html, data, or bean
undo.count

is the number of updates up to which the user can roll back a channel. When the channel is updated, the tuner keeps a record of the changes so that it can undo the channel update if needed. For example, if you set this property to 2, the user can roll back to the last two channel updates. (Channel Manager has an Undo updates option when you right-click a channel.) This property is useful if an updated channel doesn't run correctly or if the update has other problems. This property is not supported for channels created using Application Packager.

Valid value: an integer (0 through 25)

update.action

is the action the tuner performs when the channel is running and new data is available. Deprecated.

Valid value: ignore (tuner does nothing), restart (restarts the channel), or install (automatically installs new data).

update.active

is the schedule at which the channel is updated when it is running. This property is used only if update.schedule is not set. Deprecated. Provides backward compatibility update schedule for active channels.

Valid value: see “Syntax for the schedule string” on page 259.

update.check

has, when set to true, the scheduler checking for updates instead of actually performing the updates. When set to false, the scheduler performs the updates.

Valid value: true or false

update.inactive

provides a backward compatibility update schedule for inactive channels. It is used only if update.schedule is not set. Deprecated.

Valid value: see “Syntax for the schedule string” on page 259.

update.schedule

is the schedule for a channel update.

Valid value: see “Syntax for the schedule string” on page 259.

update.vary

has, when set to true, the scheduler varying the scheduled update time slightly with a random number determined by the value of update.vary.max. The random value is generated when the tuner starts and does not change until the tuner is restarted or the schedule is changed. This property is useful for spreading network traffic over time.

Valid value: true or false

update.vary.max

specifies, when update.vary is set to true, the maximum number of minutes that the next update varies from the scheduled update time. For example, 100 indicates that the actual update can occur at any point up to 100 minutes after the scheduled update time. When update.vary is set to true and this property is not set, the value used is not less than 2 and not more than approximately half the scheduled update interval.

Valid value: an integer

width

is, for applet channels, the width of an applet in pixels. AppletPanel uses this property.

Valid value: an integer

windows.icon

is the path of the Windows BMP file to use when creating the start menu or a shortcut. If not set, the Channel Manager attempts to use thumbnail or icon.smaller.

Valid value: the path name

Syntax for the schedule string

A schedule string can have the following forms:

- every <number> days update <when>
- every <number> weeks on <day_of_week> update <when>
- weekdays update <when>
- never

where:

- <number> is an integer. Even if the number is 1, the word following it in the syntax must remain plural (for example: 1 days, not 1 day).

- <day_of_week> is mon, tue, wed, thu, fri, sat, or sun (lowercase and with no punctuation). To specify more than one day of the week, use a plus sign (for example: mon+tue+fri).
- <when> is the exact time of day or time interval at which to perform an update. It can be either of the following:
 - at <time>
 - every <number> minutes | hours [between <time> and <time>] where <time> is <hour>:<minute>{am|pm} (with no space before am or pm). Precede single-digit hours with a zero (for example: 01:30pm). For midnight, use 12:00am, and for noon, use 12:00pm. If you don't specify a between interval, then all day (between 12:00am and 11:59pm) is assumed.

Examples of schedule strings follow:

- every 2 days update at 12:00am
- every 3 weeks on mon+sat update every 4 hours between 06:00am and 10:00pm
- every 2 weeks on mon update at 04:00am

These examples apply to updating schedules, but the schedule string also applies to starting schedules for channels.

Segment platform codes

The following table lists segment platform codes.

Table 5-1: Segment platform codes

Platform codes	Operating system/CPU
any	any/any
AIX,POWER_PC	IBM AIX.*/PowerPC
AIX,POWER_PC64	IBM AIX.*/PowerPC64
HP-UX,IA64	Hewlett-Packard HP-UX.*/IA64
HP-UX,PA-RISC	Hewlett-Packard HP-UX.*/PA-RISC
HP-UX,PA-RISC2.0	Hewlett-Packard HP-UX.*/PA-RISC2.0
Linux,i386	Red Hat Linux.*/i386
Macintosh,PowerPC	Apple Macintosh or MacOS/PowerPC
Mac_PowerPC,PowerPC	Apple Macintosh PowerPC or Mac OS/PowerPC
Solaris,sparc	Solaris/SPARC or SunOS/SPARC

Table 5-1: Segment platform codes (Continued)

Platform codes	Operating system/CPU
Solaris,x86	Solaris/x86
Windows CE,StrongARM	Microsoft Windows CE/StrongARM
Windows CE, Intel 486	Microsoft Windows CE/Intel486
Windows XP,x86	Microsoft Windows XP/x86
Windows,x86	Microsoft Windows generic Win32 platform/x86
Windows 2000,x86	Microsoft Windows 2000/x86
Windows 2003,x86	Microsoft Windows 2003/x86
Windows Vista	Windows Vista/x86
Windows 2008	Windows 2008/x86
Windows 7	Windows 7/x86

Segment locale codes

The following table lists the segment locale codes, based on MIME format.

Table 5-2: Segment locale codes

Locale code	Language/Location
any	any/any
da_DK	Danish/Denmark
de_AT	German/Austria
de_CH	German/Switzerland
de_DE	German/Germany
el_GR	Greek/Greece
en_CA	English/Canada
en_GB	English/United Kingdom
en_IE	English/Ireland
en_US	English/United States
es_ES	Spanish/Spain

Table 5-2: Segment locale codes (Continued)

Locale code	Language/Location
fi_FI	Finnish/Finland
fr_BE	French/Belgium
fr_CA	French/Canada
fr_CH	French/Switzerland
fr_FR	French/France
it_CH	Italian/Switzerland
it_IT	Italian/Italy
ja_JP	Japanese/Japan
ko_KO	Korean/Korea
nl_BE	Dutch/Belgium
nl_NL	Dutch/Netherlands
no_NO	Norwegian(Nynorsk)/Norway
no_NO_B	Norwegian(Bokm)/Norway
pl_PL	Polish/Poland
pt_BR	Portuguese/Brazil
pt_PT	Portuguese/Portugal
sv_SE	Swedish/Sweden
th_TH	Thai/Thailand
tr_TR	Turkish/Turkey
zh_CN	Chinese(Simplified)/China
zh_TW	Chinese(Traditional)/Taiwan

Properties for Application Packager channels

This section lists properties that are specific to channels created using Application Packager.

`adapter.state`

is the state of the adapter.

Valid value: a string

adapter.version

is the major and minor version of the adapter.

Valid value: a string

channelmanager.rightclickmenuprefix

is the right-click menu options of the package in Channel Manager.

Specify the string that you want to appear in Channel Manager (for example, Install), enter a comma (,), and then specify the command-line option you want to associate with that string (for example, -install). If you want to specify multiple strings, separate them using commas. For example,

```
channelmanager.rightclickmenuprefix=Install,-install,Uninstall  
,-remove,Verify,-verify -setpropshowfail\=true,Repair,-repair  
-setpropshowfail\=true
```

Valid value: a string

channelmanager.status.subscribed

is the state of the package on the tuner.

Valid value: a string

channeltype

is the type of channel. This property enables the transmitter to identify the channel. The channel type is written in the properties.txt file of the package directory when the channel is created.

Valid value: ApplicationPackage.shrinkwrap (shrinkwrap package),
ApplicationPackage.Generic (custom package),
ApplicationPackage.MSI (Windows package),
ApplicationPackage.FilePak (file package),
ApplicationPackage.DotNet (.NET package),
ApplicationPackage.JavaPak (Java package),
ApplicationPackage.pda (PDA package)

dsl.manufacturer

specifies the normalized manufacturer name for all channel types accept the Java package. Manufacturer name data in the Definitive Software Library (DSL) is consistently denoted with this setting.

Valid value: a string

`dsl.product`

specifies the normalized product name for all channel types accept the Java package. Product name data in the Definitive Software Library is consistently denoted with this setting.

Valid value: a string

`dsl.version`

specifies the normalized version for all channel types accept the Java package. Version data in the Definitive Software Library is consistently denoted with this setting.

Valid value: a string

`dsl.patchlastbuildid`

specifies the normalized last patch build ID for all channel types accept the Java package. Last patch build ID data in the Definitive Software Library is consistently denoted with this setting.

Valid value: an integer

`dsl.description`

specifies the normalized package description for all channel types accept the Java package. Package description data in the Definitive Software Library is consistently denoted with this setting.

Valid value: a string

`dsl.addtods1`

specifies whether the package is added to the Definitive Software Library.

Valid value: true or false

`edit.time`

is the date-time stamp of the last edit of the package.

Valid value: a long integer

`logs.enabled`

enables adapter logging when set to true.

Valid value: true or false

Default value: true

`logs.roll.policy`

is the adapter's log rolling policy: `bysize` rolls after a specified amount of disk space and `other` rolls after a specified amount of time.

Valid value: `bysize` or `other`

Default value: `bysize`

`logs.roll.size`

is the log roll size in kilobytes (Kb). This property is valid only if the `logs.roll.policy` property is set to `bysize`.

Valid value: an integer

Default value: 32

`logs.roll.versions`

is the number of previously rolled log files that can exist for the channel.

Valid value: an integer

Default value: 1

`schedule.args`

is the schedule that is used to verify and repair the package.

Valid value: `-verify`, `-repair`, or `other`

Default value: `null`

`segmentation.hierarchy`

specifies the segmentation hierarchy of the segment which is downloaded on the client.

Default value: `default`

`title`

is the title of the package.

Valid value: a string

Default value: an empty string

Properties for the Policy Manager channel

This section lists properties that are specific to the Policy Manager channel.

You can use these properties when the Policy Service and Patch Service channels on the endpoints are from a location other than the master transmitter. You use these properties to specify the channel URLs to use for these service channels so that Policy Manager does not download these channels from the master transmitter. Specify these properties in the `properties.txt` file in the `data/persist` directory of the Policy Manager channel's directory (for example, `ch.4/data/persist/properties.txt`). Restart Policy Manager after specifying the properties.

`subscriptionmanager.push.patchserviceurl`

is the channel URL to use for the Patch Service channel. This property was added in version 6.0.2.1.

Valid value: the channel URL

`subscriptionmanager.push.subscriptionserviceurl`

is the channel URL to use for the Policy Service channel. This property was added in version 6.0.2.1.

Valid value: the channel URL

Properties for the Tuner Update Manager channel

This section lists the properties that are specific to the Tuner Update Manager channel.

You set these properties in the `application.txt` file of the channel.

Note: The Tuner Update Manager channel is no longer used for upgrading to tuners that are version 6.0 and higher.

receiveBPS

is the rate of download in bytes per second for the Tuner Update Manager channel. The Tuner Update Manager channel locates this channel property in the `application.txt` file, and, if this channel property is not set, the channel locates the tuner property `marimba.bandwidth.max`. Setting this property provides flexibility to control and manage tuner updates over slow and non-dedicated network connections. This property was added in version 4.6.2.2.

Valid value: an integer

This property is deprecated.

restartTuner

indicates whether the Tuner Update Manager automatically restarts the tuner or defers the restart until after the tuner update is successfully downloaded. When set to `false`, the tuner does not restart after a tuner update. This property was added in version 4.6.2.2.

Valid value: `true` or `false`

Default value: `true`

This property is deprecated.

Chapter

6 Parameters

You can use channel parameters to control how the packaged application is installed and launched, and the Schema Manager parameters when setting up an automated installation.

The following topics are provided:

- Channel parameters (page 270)
- Schema Manager parameters (page 299)

Channel parameters

When you package an application using Application Packager, you can use *channel parameters* to control how the packaged application is installed and launched. These channel parameters are stored in the `parameters.txt` file in the channel's directory. This section lists the different channel parameters that you can specify for a packaged application.

You can edit the channel parameters for a packaged application by:

- Using Channel Copier to edit the channel parameters when you publish the channel.
- Editing the `parameters.txt` file directly. You can find the file in the *package directory*, the directory in which you instructed Application Packager to store the files for a packaged application.

Some of these channel parameters have default values. Some are additional parameters that are not found in the `parameters.txt` file by default. Entries in the `parameters.txt` file have the form:

`<channel_parameter>=<parameter_value>`

For example:

`silent=true`

Note: You can change some channel parameters through the graphical user interface (GUI), while Marimba products set other parameters, and changing them manually is not recommended.

Note: You can use some of these channel parameters only if you packaged the channel using a certain version of Application Packager. This version is specified in the following partial list of channel parameters.

`adapter.debug`

specifies whether the output that appears in the Application Installer console is verbose. Verbose output is helpful when debugging and troubleshooting packaged applications.

Valid value: `true` (verbose) or `false` (not verbose)

Default value: `false`

Application Packager version: 4.0 and later

`adapter.updateinstall`

specifies whether the channel automatically installs itself after an update.

Valid value: `true` (installed) or `false` (not installed)

Default value: `true`

Application Packager version: 4.6.1 and later

`adapter.updateinstall.pending`

specifies whether install-pending channels automatically install updates.

Valid value: `true` (installed) or `false` (not installed)

Default value: `true`

Application Packager version: 4.7.0.3 and later

`adapter.updateinstall.silent`

determines the installation mode. If set to `true`, an update that is automatically set to install (`adapter.updateinstall` is set to `true`) does so in silent mode. If set to `false`, updates have the same installation mode as the one set during packaging (silent, semisilent, or full GUI mode).

If full GUI mode was specified during packaging, the following parameters are set:

- `silent=false`
- `semisilent=false`

For updates to install in silent mode, the following parameters must be set:

- `adapter.updateinstall.silent=true`
- `silent=true`
- `semisilent=false`

Valid value: `true` or `false`

Default value: `true`

Application Packager version: 4.7 and later

`adapter.update.minor.post.repair`

specifies, when a minor update occurs, whether a repair operation is performed after the minor update is completed. This parameter does not cause any pre-repair or post-repair scripts to run.

Valid value: true (performed) or false (not performed)

Default value: false

Application Packager version: 4.7.3 and later

adapter.update.post.repair

specifies, when a major update occurs, whether a repair operation is performed after the content of the major update is installed. This parameter does not cause any pre-repair or post-repair scripts to run.

Valid value: true (performed) or false (not performed)

Default value: false

Application Packager version: 4.7.3 and later

allowrunexe

specifies whether the -runexe option can execute any program from the command line.

Valid value: true (can execute) or false (cannot execute)

Default value: false

delayfiledownload

determines whether the download has a delay. If set to true, the download follows a two-step process: The manifest and the Application Installer are downloaded first; the files in the channel are downloaded later. This parameter is useful when used with the preload parameter, pre-install scripts, or dependencies.

Setting `preload` to true automatically implies that `delayfiledownload` is also set to true; the parameter does not need to be set separately.

Valid value: true or false

Default value: false

Application Packager version: 4.5 and later

dependChannelSilent

determines how required channels are installed, after you configure the channel dependencies. If set to true, the required channels are subscribed and installed in silent mode. If set to false, the required channels are subscribed and installed using the installation options that you set using the Package Editor.

Valid value: true or false

Default value: true

Application Packager version: 4.6.2 and later

`depends.update`

specifies whether the package dependencies are executed during the pre-update phase.

Valid value: true (executed) or false (not executed)

Default value: false

`dialog.timeouts`

lets you automate full GUI or semisilent installations without user interaction; if there is no user response to the dialog box within the timeout period, the default for that dialog box is used. If set to true, this parameter makes the adapter set a timeout of 60 seconds for reboot dialog boxes and 30 seconds for other dialog boxes.

Valid value: true or false

Default value: false

`disableProcessRedirect`

specifies whether to enable or disable the process output redirection of launched scripts and executable files. Redirection of output is potentially a problem with a few Windows applications because it causes them to stop responding.

Valid value: true (disable) or false (enable)

Default value: false

Application Packager version: 4.5.1 and later

`dlltimeout`

sets the maximum timeout duration for registering a dll. Application Packager terminates the dll registration if it fails to complete within the specified duration and logs this event as "DLL is not registered. Dlltimeout value elapsed: -3", in the channel history.

`editor.save.history.prompt.suppress`

determines whether a prompt appears. If set to false, specifies that the prompt appears, asking users to add a history entry to the package before it is saved. If set to true, the prompt does not appear. This parameter is a package time property and has no effect at install time.

Valid value: true or false

Default value: false

Application Packager version: 4.7.2 and later

fileswithchannel

enables or disables the UpdateNow SDK for versions earlier than 4.5. If set to true, the files are sent down as part of the channel. Duplicate storage (channel files are stored both in the tuner's workspace directory and in the file system) is a side-effect.

For version 4.5 or later, use the `nofilemap` parameter instead.

Valid value: true (enables) or false (disables)

Default value: false

forceterm

specifies whether to force the launched application to terminate after a tuner exit is requested. This parameter is valid only if the `rundetach` parameter is set to false. You can specify how long to wait before terminating the application using the `forcetimeout` parameter.

Valid value: true or false

Default value: false

Application Packager version: 4.5 and later

forcetimeout

specifies the number of seconds to wait before forcefully terminating the launched application after a tuner exit is requested. This parameter is useful when you want to allow applications to exit gracefully, or when you want to allow users to save their work before forcing applications to exit.

This parameter is valid only if the `rundetach` parameter is set to false, and the `forceterm` parameter is set to true.

Valid value: an integer (in seconds)

Default value: 20

Application Packager version: 4.5 and later

inifileobjects

specifies how Application Packager processes .ini files. If set to true, .ini files are treated as special .ini file objects and Application Packager merges found changes with the existing file found at the endpoint. If set to false, Application Packager processes .ini files as ordinary files and does not merge changes.

For channels packaged using the Custom Application Packager and for editing previously packaged channels, you can access the channel directory (the directory in which you instructed Application Packager to store the files for a packaged application before publishing) and set the inifileobjects parameter in the parameters.txt file.

If you are going to use Packager for Shrinkwrap Windows Applications and File Packager, the channel directories are usually not yet created. Set the inifileobjects parameter in the parameters.txt file in the Application Packager channel directory in the tuner workspace (on Windows, the path is similar to C:\winnt\.marimba\<tuner_keyword>\ch.3\data\parameters.txt). Setting the inifileobjects parameter here allows all .ini files that the Application Packager encounters during the packaging process to be treated as ordinary files.

Valid value: true or false

Default value: true

Application Packager version: 4.6.2 and later

install.priority

specifies the priority of the Application Installer thread performing the installation.

Valid value: an integer

Default value: default Java thread priority

Application Packager version: 4.0.2 and later

msi.commandline.<mode>

specifies, if msi.commandline.<mode>.use is set to true, what to execute in the given mode, where <mode> is one of the following strings:

- install
- update.minor
- update.major

- verify
- repair
- uninstall

This parameter is valid for MSI packages only.

Valid value: a shell command line

Default value: null

`msi.commandline.<mode>.launchflags`

specifies the process creation flags for the command line specified by `msi.commandline.<mode>`, where `<mode>` is one of the modes listed for `msi.commandline.<mode>`. These flags determine how the process for the command line is created.

This parameter is valid for MSI packages only.

Valid value: See the values listed for `processCreationFlags`.

`msi.commandline.<mode>.use`

determines whether the command line specified by `msi.commandline.<mode>` is executed instead of the MSI APIs.

This parameter is valid for MSI packages only. `<mode>` is one of the modes listed for `msi.commandline.<mode>`.

Valid value: true (`msi.commandline.<mode>`) or false (MSI APIs)

Default value: false

`msi.db`

specifies the file name of the .msi file in an MSI package.

This parameter is valid for MSI packages only.

Valid value: a file name

Default value: null

`msi.download`

specifies the file retrieval mode for an MSI installation.

This parameter is valid for MSI packages only.

Valid value:

- all—Download all files.
- required—Download only required files.

- `msi` – Download .msi file only.

Default value: `all`

`msi.download.delete`

determines whether, after the MSI package has been successfully installed, downloaded files are deleted from the local system.

This parameter is valid for MSI packages only.

Valid value: `true` (deleted) or `false` (not deleted)

Default value: `false`

`msi.download.path`

specifies the URLs and paths from which to retrieve files if they are needed by the Windows Installer for additional feature installation or repair.

This parameter is valid for MSI packages only.

Valid value: semicolon-separated list of URLs, and local and network paths

Default value: an empty string

`msi.install`

specifies the mode of installation for an MSI package.

This parameter is valid for MSI packages only.

Valid value:

- `default`–Perform normal application install.
- `admin`–Perform administrative install (for redistribution).
- `asis`–Install packaged files “as is.”

Default value: `default`

`msi.install.args`

specifies the command-line property settings to use during installation (in `property=value` form, each separated by at least one space).

This parameter is valid for MSI packages only.

Valid value: a string

Default value: an empty string

`msi.installed.reinstall`

determines whether the application is reinstalled if it was previously installed prior to first Marimba deployment. This parameter is valid for MSI packages only.

Valid value: `true` (reinstalled) or `false` (not reinstalled)

Default value: `false`

`msi.log.attr`

specifies the flags to control the MSI logging attributes.

This parameter is valid for MSI packages only.

Valid value: an integer

Default value: 0

`msi.log.mode`

specifies the flags to control the MSI logging modes.

This parameter is valid for MSI packages only.

Valid value: an integer

Default value: 0

`msi.log.path`

specifies the user-controlled MSI log file path.

This parameter is valid for MSI packages only.

Valid value: a file path

Default value: an empty string

`msi.patch.change.reinstall`

determines whether the application is reinstalled so changes to the patch list are applied.

This parameter is valid for MSI packages only.

Valid value: `true` (reinstalled) or `false` (not reinstalled)

Default value: `false`

`msi.policy.elevated`

determines whether Windows Installer installs with elevated privileges.

This parameter is valid for MSI packages only.

Valid value: true (elevated) or false (not elevated)

Default value: non-existent

`msi.policy.enabled`

specifies when to enable or disable Windows Installer.

This parameter is valid for MSI packages only.

Valid value:

- empty string—Use the system's default setting.
- 1—(Managed) Enable for managed applications only.
- 0—(All) Enable for all applications.

Default value: empty string

`msi.policy.ldbrowse`

determines whether browsing by non-admin users is allowed or disallowed.

This parameter is valid for MSI packages only.

Valid value: true (allowed) or false (disallowed)

Default value: non-existent

`msi.policy.log`

specifies whether or not Windows Installer uses default MSI logging.

This parameter is valid for MSI packages only.

Valid value:

- non-existent—Use the default setting for MSI default logging.
- non-empty string—Allow MSI default logging.
- empty-string—Disallow MSI default logging.

Default value: non-existent

`msi.policy.nobrowse`

determines whether browsing for alternative install sources in an MSI package is allowed or disallowed.

This parameter is valid for MSI packages only.

Valid value: true (disallowed) or false (allowed)

Default value: `false`

`msi.policy.nomedia`

determines whether the use of alternate media sources is allowed or disallowed.

This parameter is valid for MSI packages only.

Valid value: `true` (disallowed) or `false` (allowed)

Default value: non-existent

`msi.policy.norollback`

determines whether rollback files are stored during install.

This parameter is valid for MSI packages only.

Valid value: `true` (not stored) or `false` (stored)

Default value: non-existent

`msi.product`

specifies the MSI application product code.

This parameter is valid for MSI packages only.

Valid value: a string

Default value: an empty string

`msi.product.advertise`

specifies the advertised state. If set to `true`, this parameter specifies that the MSI product to be installed is advertised. If this parameter is toggling from `true` to `false`, the MSI product on the endpoint is installed if it is in the advertised state.

This parameter is valid for MSI packages only.

Valid value: `true` or `false`

Default value: non-existent

Application Packager version: 4.7.2 and later

`msi.product.installed`

specifies the installed state. If set to `true`, this parameter specifies that the MSI product is installed. If set to `false`, this parameter specifies that the MSI product is not installed.

This parameter is valid for MSI packages only.

Valid value: true or false

Default value: non-existent

Application Packager version: 4.7.2 and later

msi.redirect

determines whether the MSI channel can download contents from repeaters that support MSI redirection, if there are any available. This parameter is used in conjunction with the download MSI only mode.

This parameter is valid for MSI packages only.

Valid value: true (can download) or false (cannot download)

Default value: false

Application Packager version: 4.7.2 and later

msi.repair.mode

specifies the list of MSI repair options used when performing a repair. These repair options correspond to the MSI command-line options. For more information, see the Microsoft Windows Installer documentation or the Microsoft website (<http://msdn.microsoft.com>).

For example:

```
msi.repair.mode=filemissing,fileequalversion,fileverify,  
machinedata,shortcut
```

This parameter is valid for MSI packages only.

Valid value: A comma-separated list of MSI repair options, including:

- filemissing—Reinstall only if the file is missing.
- fileoldversion—Reinstall if the file is missing or an older version is installed.
- fileequalversion—Reinstall if the file is missing or an equal or older version is installed.
- fileexact—Reinstall if the file is missing or a different version is installed.
- fileverify—Reinstall if the file is missing or the stored checksum doesn't match the calculated value (the file might be corrupted).
- filereplace—Reinstall all files, regardless of version.
- machinedata—Rewrite all required computer-specific registry entries.

- **userdata**—Rewrite all required user-specific registry entries.
- **shortcut**—Overwrite all existing shortcuts.
- **package**—Run from source and re-cache the local package.

Default value: empty string

Application Packager version: 4.7.0.3 and later

`msi.repair.source.download`

determines whether a repair of the cached MSI source files in the tuner workspace directory must be completed before performing an MSI repair operation.

Valid value: `true` (must be completed) or `false` (does not have to be completed)

Default value: `false`

Application Packager version: 4.7.3 and later

`msi.source.alternate`

determines whether an MSI installation or repair uses this source to download content. If `msi.redirect` is set to `true`, this alternate path overrides any source the channel is set to download from.

This parameter is valid for MSI packages only.

Valid value: an HTTP URL path or network path in the Universal Naming Convention (UNC) format (for example,

`\\\\servername\\\\sharename\\\\path\\\\directory_containing_MSU_file`

or

`\\\\ipaddress\\\\sharename\\\\path\\\\directory_containing_MSU_file`)

Default value: `null` (does not use this source)

Application Packager version: 4.7.2 and later

`msi.transform.change.reinstall`

determines whether the application is reinstalled so changes to the transform list are applied.

This parameter is valid for MSI packages only.

Valid value: `true` (reinstalled) or `false` (not reinstalled)

Default value: `false`

msi.transform.path

specifies the list of paths of additional transforms (on the target system) to be applied to the MSI installation.

This parameter is valid for MSI packages only.

Valid value: semicolon-separated list of paths

Default value: empty string

msi.ui

specifies the UI level for an MSI package.

This parameter is valid for MSI packages only.

Valid value:

- default—The Windows Installer chooses its default UI level.
- none—Completely silent installation.
- basic—Simple progress and error handling dialogs are shown.
- reduced—Authored dialogs are suppressed.
- full—All authored, progress, and error dialogs are shown.

Default value: default

msi.ui.basic.progressonly

specifies that the UI level for an MSI installation is `basic` as specified in the `msi.ui` parameter, but eliminates any modal dialog boxes and shows progress bars only. If there are errors, warning pop-up windows are not shown to users.

This parameter works only if you used the Package Editor to set the UI level to `basic` in the Windows Installer – UI Level page (in the `parameters.txt` file, `msi.ui` is set to `basic`).

This parameter is valid for MSI packages only.

Valid value: true or false

Default value: false

Application Packager version: 4.6.1 and later

msi.ui.flags

specifies the level of complexity of the user interface for an MSI package.

This parameter is valid for MSI packages only.

Valid value: a comma-separated list of the following constants:

- INSTALLUILEVEL_NOCHANGE
- INSTALLUILEVEL_DEFAULT
- INSTALLUILEVEL_NONE
- INSTALLUILEVEL_BASIC
- INSTALLUILEVEL_REDUCED
- INSTALLUILEVEL_FULL
- INSTALLUILEVEL_ENDDIALOG
- INSTALLUILEVEL_PROGRESSONLY
- INSTALLUILEVEL_HIDECANCEL
- INSTALLUILEVEL_SOURCERESONLY

For descriptions about each of these constants, see the MSI documentation, or refer to the Microsoft website (<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/msi/setup/msisetinternalui.asp>).

Default value: null

`msi.ui.raw`

adds additional MSI user interface options for an MSI package. This parameter can be useful if new user interface options become available for MSI, and the Application Packager does not yet have GUI support for them.

Make sure the MSI package has the exact constants or integer values, which can be found in the `msi.h` header file in the Windows Installer SDK. This parameter is ORed together with the `msi.ui.flags` parameter.

This parameter is valid for MSI packages only.

Valid value: an integer

Default value: null

`msi.upgrade.uninstall`

determines whether the MSI application is uninstalled if the MSI database has changed after an update. If the MSI application is uninstalled, it is then reinstalled.

This parameter is valid for MSI packages only.

Valid value: true (uninstalled) or false (not uninstalled)

Default value: false

`msi.upgrade.uninstall.checkresult`

specifies if the return code for uninstalling the MSI application is relevant when determining whether the MSI channel has failed to update. This parameter is checked only if the `msi.upgrade.uninstall` parameter is true.

This parameter is valid for MSI packages only.

Valid value: `true` (is relevant) or `false` (is not relevant)

Default value: `true`

`nobackup`

specifies if you want to store files and registry entries for backup during an installation.

Usually, when a channel is installed, existing files and registry entries that are deleted or overwritten are stored in a backup directory (the `unfiles` directory of the channel in the tuner's workspace directory). These files are used to restore the system to its previous state when the channel is uninstalled. This parameter can be useful if you want to make sure that an application does not reside on endpoints after a channel is uninstalled. It also allows you to save some disk space.

WARNING: Selecting this parameter can cause the removal of files and registry entries that are critical to other applications.

Valid value: `true` (do not store) or `false` (store)

Default value: `false`

Application Packager version: 4.7 and later

`nofilemap`

determines whether file mapping is used when copying files to the file system. If set to `true`, when a channel is installed, channel files are stored both in the tuner's workspace directory and in the file system.

This parameter is useful if you want to duplicate the storage of channel files. For example, you want to check the verify and repair operations even when a computer is not connected to the network.

Note: This parameter should not be used with `preload=true`.

Valid value: true (not used) or false (used)

Default value: false

Application Packager version: 4.5 and later

`nostoragecompact`

determines whether the Application Installer compacts the storage periodically during installation. Not compacting can improve the installation performance of large channels.

Valid value: true (does not compact) or false (compacts)

Default value: false

Application Packager version: 4.5 and later

`notifyremove`

determines whether to notify the Application Installer that a channel is being deleted. The deletion triggers an uninstall of the files in the channel.

Valid value: true (notify) or false (do not notify)

Default value: true

Application Packager version: 4.5 and later

`noverifychecksums`

determines whether the Application Installer tries to verify the checksums of the files while installing them. Not verifying can improve the installation performance of large channels.

Valid value: true (does not verify) or false (verifies)

Default value: false

Application Packager version: 4.0.2 and later

`noverifyspace`

determines whether to disable checking of sufficient disk space before starting installation. Set this parameter to true to get around a defect in some mounted drives, which incorrectly report the amount of free disk space as always 0.

Valid value: true (disables) or false (does not disable)

Default value: true

Application Packager version: 4.5 and later

patch.args.<string>

(where <string> is a name)

specifies the arguments for an MSI patch with a given name.

This parameter is valid for MSI packages only.

Valid value: a string

Default value: an empty string

patch.count

specifies the number of patches in the MSI package.

This parameter is valid for MSI packages only.

Valid value: integer

Default value: 0

patch.<x>, where <x> is an integer

specifies an MSI patch name.

This parameter is valid for MSI packages only.

Valid value: a string

Default value: null

postscripts.fail

determines whether the failure of post-operation scripts (such as post-install scripts) causes the package operation to fail. The package operation fails if the post-operation script returns a non-zero value.

Before version 4.7.2.1, post-operation scripts had no effect on whether the package operation succeeded or failed. Specifically, post-install, post-major update, post-minor update, post-verify, post-repair, and post-uninstall scripts did not have any effect on whether or not the operation succeeded. To preserve this behavior in 4.7.2.1 and later, set this parameter to false.

Valid value: true (fails) or false (does not fail)

Default value: true

Application Packager version: 4.7.2.1 and later

preload

determines whether the Application Installer performs an extra step before it downloads and installs files on the user's system.

If set to true, the Application Installer scans the channel's manifest for files that already exist on the user's system, and then adds those files to the channel's index. This lets you save time by not having to download files already on the user's system. However, this may cost extra CPU time during installation and may make the overall installation take longer.

Setting this parameter to true automatically implies that the `delayfiledownload` parameter is also set to true. You do not need to set the `delayfiledownload` parameter separately.

Note: This parameter should not be used with `nofilemap=true`.

Valid value: true (extra step) or false (no extra step)

Default value: false

Application Packager version: 4.0 and later

processCreationFlags

controls process creation on Windows platforms when processes are created using the following methods:

- From the “Launch application from tuner” option under Configuration > Startup in the Package Editor
- From executable script files (with the extensions .exe or .bat)
- From the `-runexe` command-line option for packaged applications

Use a comma-separated list if using more than one value. For more information about these keywords, search for “process creation flags” on the [Microsoft website \(<http://msdn.microsoft.com>\)](http://msdn.microsoft.com).

Valid value:

- DEBUG_PROCESS
- DEBUG_ONLY_THIS_PROCESS
- CREATE_SUSPENDED
- DETACHED_PROCESS
- CREATE_NEW_CONSOLE
- NORMAL_PRIORITY_CLASS
- IDLE_PRIORITY_CLASS
- HIGH_PRIORITY_CLASS
- REALTIME_PRIORITY_CLASS

- CREATE_NEW_PROCESS_GROUP
- CREATE_UNICODE_ENVIRONMENT
- CREATE_SEPARATE_WOW_VDM
- CREATE_SHARED_WOW_VDM
- CREATE_FORCEDOS
- BELOW_NORMAL_PRIORITY_CLASS
- ABOVE_NORMAL_PRIORITY_CLASS
- CREATE_BREAKAWAY_FROM_JOB
- CREATE_DEFAULT_ERROR_MODE
- CREATE_NO_WINDOW

queryCheckUpdate

determines whether the channel checks if an update is available on the transmitter when the channel is launched to run an application. If a channel update is available, the channel queries the user about downloading and installing the update. If the user chooses not to download and install the update, the application is run.

Valid value: true (checks) or false (does not check)

Default value: false

Application Packager version: 4.6 and later

reboot.allow

determines whether the channel can reboot the machine if required.

Note: Channels do not cause machines to reboot automatically in silent mode, unless the `reboot.force` parameter is set to true.

For information about how this parameter works with the Policy Service channel, see the *Policy Management User Guide*, available on the Marimba Channel Store.

Valid value: true (can reboot) or false (cannot reboot)

Default value: true

Application Packager version: 4.5 and later

reboot.allowcancel

determines whether you can cancel a reboot. If set to true, and a reboot is required on the endpoint, then a reboot option letting you cancel the reboot appears in the reboot dialog box (only in full UI or semi-silent UI). If the installer is in silent UI, then no reboot is done, even if one is required.

Valid value: true (can cancel reboot) or false (cannot cancel reboot)

Default value: true

Application Packager version: 4.7.1 and later

reboot.force

determines whether a machine reboots after an installation. If set to true, unwanted user interaction can result.

For information about how this parameter works with the Policy Service channel, see the *Policy Management User Guide*, available on the Marimba Channel Store.

Valid value: true (reboots) or false (does not reboot)

Default value: false

Application Packager version: 4.5 and later

reboot.showdialog

indicates that, in interactive mode (not silent or semisilent mode), the channel shows the reboot dialog box as necessary. To override this behavior, set the parameter to false.

For information about how this parameter works with the Policy Service channel, see the *Policy Management User Guide*, available on the Marimba Channel Store.

Valid value: true or false

Default value: true

Application Packager version: 4.5 and later

reboot.wait.seconds

controls how many seconds to wait before rebooting. Increasing the wait time allows channel states to be flushed to disk before rebooting on busy machines.

Valid value: integer (in seconds)

Default value: 4

Application Packager version: 4.7.2.1 and later

rollback

specifies if you want the channel to automatically roll back to the previous state if installation fails.

If installation fails when a channel is being installed for the first time, it automatically rolls back the channel to the uninstalled state. Files and other items in the channel that were installed are uninstalled, while items that were deleted or modified during installation are restored.

If an update to a channel fails, it automatically rolls back the channel to its state before the update.

Valid value: true or false

Default value: true

Application Packager version: 4.7 and later

run

specifies the executable to run when the channel is launched. It can include a macro, such as \$worddir\word.exe.

Valid value: path and name of the executable

Default value: empty string

Application Packager version: 4.0 and later

runargs

specifies arguments to pass to the executable (specified by the run parameter). It can include macros.

Valid value: arguments

Default value: empty string

Application Packager version: 4.0 and later

rundetach

determines whether the launched application runs independently from the tuner.

Valid value: true (runs independently) or false (does not run independently)

Default value: false

Application Packager version: 4.5 and later

rundir

specifies the current working directory to use when running the executable specified by the run parameter. Like the run parameter, it can include a macro.

Valid value: path of the current working directory

Default value: empty string

Application Packager version: 4.0 and later

run.install

determines whether an application starts when a channel is installed for the first time, if you have previously specified the application to start when the channel is launched (by either specifying the executable file in the Package Editor or the `run` parameter).

Valid value: `true` (starts) or `false` (does not start)

Default value: `false`

Application Packager version: 4.6.1 and later

run.update

determines whether an application starts when a channel is updated (both minor and major updates), if you have specified the application to start when the channel is launched (by either specifying the executable file in the Package Editor or the `run` parameter).

Valid value: `true` (starts) or `false` (does not start)

Default value: `false`

Application Packager version: 4.7.2 and later

runuser

determines whether any executable launched through the channel (whether by specifying the launch path of a channel or using the command-line option `-runexe`) runs under the currently logged-in user rather than under the system account.

This parameter is valid only on Windows NT, 2000, and XP.

Valid value: `true` (logged-in user) or `false` (system account)

Default value: `false`

savedelay

sets the time interval between saving the properties file. For example, if you set `savedelay=10`, then the Application Installer saves the properties file every 10 minutes.

On large channels, constantly saving the file can significantly affect the performance.

Valid value: integer (in minutes)

Default value: 1

Application Packager version: 4.0 and later

`scripts.perm`

On UNIX, determines whether permissions for the scripts in a package are maintained. Set this parameter in the `parameters.txt` file. If `scripts.perm` is not set, the permissions for the script are set to the default of 700.

Valid value: `true` (maintains permissions) or `false` (does not maintain permissions)

Default value: `false`

Application Packager version: 6.5 and later

`semisilent`

determines whether to enable or disable semisilent mode.

In semisilent mode, the Application Installer displays no dialog boxes to the user and uses the default channel settings during user installation.

Application Installer only displays installation progress bars.

For installation in semisilent mode, the following parameters must be set:

- `semisilent=true`
- `silent=false`

Valid value: `true` (enables) or `false` (disables)

Default value: `false`

Application Packager version: 4.0 and later

`semisilent.cancel.disable`

determines whether to enable or disable the Close (X) button at the top right corner of the Application Installer dialog box when running a semisilent installation.

Valid value: `true` (disable) or `false` (enable)

Default value: `false`

Application Packager version: 4.7.2.2c and later

setreferencetime

determines the file modification date. If set to `true`, this parameter sets the file modification dates to the file dates captured during packaging time in the manifest. If set to `false`, this parameter sets the file modification dates for files packaged by reference to the current time.

Valid value: `true` or `false`

Default value: `false`

Application Packager version: 4.5 and later

showfail

determines whether a modal error dialog box showing what failed appears, if the adapter is in semi-silent mode, and a failure during the installation occurs.

Valid value: `true` (dialog box) or `false` (no dialog box)

Default value: `false`

silent

determines whether to enable or disable silent mode.

In silent mode, the Application Installer displays no dialog boxes or progress bars to the user and uses all of the default channel settings during user installation.

For installation in silent mode, the following parameters must be set:

- `silent=true`
- `semisilent=false`

Valid value: `true` (enables) or `false` (disables)

Default value: `false`

Application Packager version: 4.0 and later

simulate

determines whether to have a preview mode in which the package goes through the act of installation without making any actual changes.

Valid value: `true` (preview mode) or `false` (no preview mode)

Default value: `false`

transform.count

specifies the number of transforms in the MSI package.

This parameter is valid for MSI packages only.

Valid value: an integer

Default value: 0

`transform.<x>`

specifies an MSI transform name, where `<x>` is an integer.

This parameter is valid for MSI packages only.

Valid value: a string

Default value: null

`uninstalllockedfile`

determines whether, for updates and uninstallations, locked files (files that are being used) that are to be removed by the Application Installer (adapter) are removed after a reboot.

Valid value: true (removed) or false (not removed)

Default value: false

`userenv`

determines whether the user environment is passed to user processes (such as scripts) that are launched from a channel created using Application Packager.

This parameter is useful when you want scripts that are set to run in the user's context to have access to the currently logged on user's environment variables. It has no effect on processes that run under the system context.

This parameter is valid only on Windows.

Valid value: true (passed) or false (not passed)

Default value: false

Application Packager version: 4.7.1.2 and later

userobjs

determines whether to prevent the installation of a package that contains user-specific content (such as registry keys and values under the HKEY_CURRENT_USER registry key, or files on the Windows desktop) if there is no user logged on to a machine.

For non-MSI packages, such as those created using Packager for Shrinkwrap Windows Applications and Custom Application Packager, setting this parameter to `true` blocks the installation of the application if it contains user-specific content, until a user logs in. This prevents the application from incorrectly installing user-specific content to the account on which the tuner service is running.

For MSI packages, this parameter indicates to the installer that it should start the MSI installation under the user's context rather than the context of the tuner service. This is useful for preventing the installation of MSI applications that install user-specific registry keys or files in user-specific directories like the Windows desktop if no one is logged onto a machine.

Note: If you edit a package and add user-specific content (such as a registry key under HKEY_CURRENT_USER or files under the macros `$SYS.AppData` and `$SYS.Desktop`), the parameter is automatically set to `true`. Setting the parameter to `true` does not occur when you add the content to a network drive or if you use a network path.

Valid value: `true` (prevents) or `false` (does not prevent)

Default value: `false`

`verifyrepair.update.install`

controls whether pending updates should be installed during a verify or repair operation. If set to `true`, the verify or repair operation installs any pending updates instead of performing a verify or repair. If set to `false`, the verify or repair operation does not attempt to install pending updates, and the verify or repair operation attempt fails.

Pending updates can exist in the following scenarios:

- The channel is installed, and the index that is cached in the channel's configuration is different from the current channel index.
- An update has been downloaded (staged), but it has not yet been installed.
- An attempt was made to install an update, but it was rolled back.

Valid value: `true` or `false`

Default value: `false`

Application Packager version: 4.7.2.1 and later

`verify.schedule`

specifies the schedule that is followed when determining if a package needs to be verified.

Valid value: a formatted string

Default value: an empty string

`wts`

specifies if a package was created on a Windows Terminal Services (WTS) machine and can be deployed to WTS machines only.

Valid value: `true` (WTS) or `false` (not WTS)

Default value: `false`

Schema Manager parameters

You can use the ADAM / AD LDS and database configuration parameters when setting up an automated installation. Specific default values listed in this section are defined in Schema Manager.

ADAM / AD LDS configuration parameters

This section lists the ADAM / AD LDS configuration parameters. ACL base DN sets the location where the access control lists (ACLs) are stored.

Valid value: a distinguished name

Default value: OU=ad,OU=Config Objects,OU=BMC CM,OU=BMC Software,DC=marimba,DC=demo Config base DN

Sets the location where Marimba objects are stored.

Valid value: a distinguished name

Default value: OU=BMC CM,OU=BMC

Software,DC=marimba,DC=demo Collections base DN

Sets the location for collections in Report Center.

Valid value: a distinguished name

Default value: OU=Collections,OU=BMC CM,OU=BMC

Software,DC=marimba,DC=demo Hidden entries

sets the list of containers that do not display on Policy Manager.

Valid value: a distinguished name

Default value: "ou=Subscriptions,ou=Config Objects,ou=BMC CM,ou=BMC Software,dc=marimba,dc=demo","ou=SpecialUsers,dc=marimba,dc=demo","ou=acl,ou=Config Objects,ou=BMC CM,ou=BMC Software,dc=marimba,dc=demo","ou=Config Objects,dc=marimba,dc=demo"

LDAP Query Collections base DN

sets the location where LDAP Query Collection objects are stored.

Valid value: a distinguished name

Default value: OU=LDAPCollections,OU=BMC CM,OU=BMC

Software,DC=marimba,DC=demo Machine import base DN

Sets the location where machine objects are stored.

Valid value: a distinguished name

Default value: OU=Machines,DC=marimba,DC=demo (for ADAM / AD LDS and Sun ONE) or OU=Computers,DC=marimba,DC=demo (for Active Directory)

schema base DN

sets the distinguished name of the schema base for creating schema attributes and classes.

Valid value: a distinguished name

Default value: DC=marimba,DC=demo Subscription base DN

sets the location where Policy Manager stores policies.

Valid value: a distinguished name

Default value: OU=Subscriptions,OU=ConfigObjects,OU=BMC CM,OU=BMC Software,DC=marimba,DC=demo

Subscription config base DN

sets the location of the Subscription Config object that stores configuration information for Policy Manager.

Valid value: a distinguished name

Default value: OU=Config Objects,DC=marimba,DC=demo

Database configuration parameters

This section lists the database configuration parameters.

`dbtree_password`

sets the password for the dbtree user.

Valid value: a string

Default value: dbtree

MRBA_XXX_growth

sets the inventory system-file growth, in MB.

Valid value: an integer in MB

Default value: 50 MB

MRBA_XXX_maxsize

sets the inventory system-file maximum size, in MB.

Valid value: an integer in MB

Default value: unlimited

MRBA_XXX_size

sets the initial data and index file sizes for schema components, in MB. For example, inventory, dbtree, or logging.

Valid value: an integer in MB

Default value: 300 MB

Chapter

7

Channel states

Channel states indicate the current condition of a channel. Usually, they also indicate the type of operation the channel has recently run. Channel states appear in the log files and allow you to monitor and troubleshoot problems with your channels.

The following topics are provided:

- Overview of channel states (page 304)
- States for Application Packager (any version) packages (page 304)
- States for Application Packager (version 4.6 or later) packages (page 305)

Overview of channel states

You can view channel states in any of the following Marimba products:

- Tuner (Channel Manager)
- Tuner Administrator
- Report Center
- Policy Management

The following channel states are common to all channels (even Marimba products that are channels):

- available
- subscribed
- unsubscribed
- running
- updating

The other channel states are only available to channels packaged using Application Packager.

States for Application Packager (any version) packages

The following table lists the channel states that are available to packages created using any version of Application Packager:

Table 7-1: States for Application Packager (any version) packages

Channel state	Explanation
available	The channel is available for download and installation.
subscribed	The channel was successfully downloaded (subscribed), but installation was not attempted.
unsubscribed	The channel was uninstalled and unsubscribed from the workspace. No updates to the channel can be downloaded.
running	The channel is running, or the channel is installing, uninstalling, verifying, repairing, or launching an application.

Table 7-1: States for Application Packager (any version) packages(Continued)

Channel state	Explanation
updating	The channel is getting updates from a transmitter.
installed	The package successfully installed an application.

States for Application Packager (version 4.6 or later) packages

The following table lists the channel states that are available for channels packaged using only Application Packager version 4.6 or later:

Table 7-2: States for Application Packager (version 4.6 or later) packages

Channel state	Explanation
install-pending	The package was successfully downloaded, and a package update was downloaded into the tuner workspace and is awaiting installation.
failed (install)	The installation of the package failed.
failed (pre-install)	A pre-installation script included in the package failed, and the installation was aborted.
failed (post-install)	A post-installation script included in the package failed. The application installed by the package may be unusable.
failed (uninstall)	The uninstallation of the package failed.
failed (pre-uninstall)	A pre-uninstallation script included in the package failed, and the uninstallation was aborted.
failed (post-uninstall)	A post-uninstallation script included in the package failed. The uninstall of the package must be attempted again to remove the package from the tuner workspace.
failed (verify)	The verification of the package failed.
failed (pre-verify)	A pre-verification script included with the package failed, and the verification was aborted.
failed (post-verify)	A post-verification script included with the package failed.
failed (launch)	The launching of the package failed. The operation was aborted.

Table 7-2: States for Application Packager (version 4.6 or later) packages (Continued)

Channel state	Explanation
failed (pre-launch)	A pre-launch script included with the package failed, and the launch was aborted.
failed (post-launch)	A post-launch script included with the package failed.
failed (repair)	The repair of the package failed.
failed (pre-update)	A pre-update script included with the package failed, and the update was aborted.
failed (post-update)	A post-update script included with the package failed.
aborted (<operation>)	<p>The Application Installer was interrupted or stopped before it could write out one of the following <operation> states:</p> <ul style="list-style-type: none"> n pre-install n install n post-install n pre-verify n verify n post-verify n pre-uninstall n uninstalled n post-uninstall n pre-update n post-update <p>The specified operation was aborted.</p>

Chapter

8

Logging codes

Each tuner maintains log files for the tuner and the channels that are on the tuner. This section lists the information that appears in the log files, including the log ID, the corresponding log message, and sometimes the severity level. You can use this information to monitor and troubleshoot problems in your Marimba system. This information is also useful with the centralized logging feature. For more information, see the *Symphony Marimba Client Automation Report Center User Guide*, which is available on the Marimba Channel Store.

The following topics are provided:

- Products without logging information (page 309)
- Log severity levels (page 309)
- Log ID ranges for specific channels (page 311)
- Action Request (page 312)
- Administration tools (page 313)
- Application Packager (page 314)
- Channel Copier (page 326)
- Common Management Services (page 328)
- Common Reboot Service (page 333)
- Content Replicator (Server Management) (page 334)
- Deployment Manager and Deployment Service (Server Management) (page 337)
- Infrastructure Service (page 363)

- Infrastructure Status Monitor (page 370)
- Inventory (page 371)
- Logging (page 377)
- Marimba Migration Module (page 379)
- Patch Management (page 381)
- Policy Management (page 387)
- Proxy (page 398)
- Proxy Administrator (page 400)
- Report Center (page 401)
- Schema Management (page 408)
- Setup and Deployment (page 410)
- Storage (page 415)
- Subnet Repeater Policy (page 416)
- Transmitter (page 417)
- Transmitter Administrator (page 425)
- Tuner (page 427)
- Tuner Administrator (page 435)
- Tuner Packager (page 439)

Products without logging information

Logging information for the following Marimba products is not available at this time:

- Certificate Manager—The tuner logs relevant certificate-related activities.
- Publisher—The transmitter that is the target of the publish operation logs relevant publishing activities.
- License Installer—The tuner logs relevant license-related activities.
- UpdateNow SDK—Not applicable. SDKs use notifications, and developers can create logs if needed.
- PublishNow SDK—Not applicable. SDKs use notifications, and developers can create logs if needed.

Log severity levels

Each log entry is assigned one of the severity levels described in the following table. The *Levels Reported* column shows which severity levels are reported when you specify a severity level for filtering log data using centralized logging in Report Center (MAJOR is the default).

Table 8-1: Log severity levels and examples

Severity level	Severity keyword	Levels reported	Examples of log IDs and messages
8	AUDIT	audit (8) info (7) critical (6) major (5) minor (4) warning (3) cleared (2)	Indicates a normal event, such as connecting to a database or installing a file. For example: 9030= Adapter finished, AUDIT [from an Application Packager channel] 9031= Script completed, AUDIT [from an Application Packager channel]
7	INFO	info (7) critical (6) major (5) minor (4) warning (3) cleared (2)	Indicates an informational message that implies no potential negative impact. For example: 1006= Kernel property changed, INFO [Tuner]

Table 8-1: Log severity levels and examples (Continued)

Severity level	Severity keyword	Levels reported	Examples of log IDs and messages
6	CRITICAL	critical (6)	<p>Indicates that a service-affecting error occurred and urgent corrective action must be taken to solve the problem. For example:</p> <p>9038= Critical dependency failed, CRITICAL [from an Application Packager channel]</p> <p>9034= Critical script failure, CRITICAL [from an Application Packager channel]</p>
5	MAJOR	critical (6) major (5)	<p>Indicates that a service-affecting error occurred and corrective action must be taken to correct the problem or to avoid a critical error. For example:</p> <p>4445= Could not connect to proxy chain URL, MAJOR [Proxy]</p> <p>9010= Operation failed, MAJOR [from an Application Packager channel]</p>
4	MINOR	critical (6) major (5) minor (4)	<p>Indicates that a non-service-affecting error occurred and corrective action should be taken to prevent the error from escalating. For example:</p> <p>1106= Channel update proxy bypassed, MINOR [Tuner]</p> <p>1720= Certificate request install failed, MINOR [Tuner]</p>
3	WARNING	critical (6) major (5) minor (4) warning (3)	<p>Indicates that a potentially service-affecting condition is detected and a diagnosis should be performed. If necessary, corrective action should be taken. For example:</p> <p>11201= No transmitter stats, WARNING [Transmitter Reporter]</p> <p>11202= No publish stats, WARNING [Transmitter Reporter]</p>
2	CLEARED	critical (6) major (5) minor (4) warning (3) cleared (2)	Indicates that a previously reported error is cleared.

Log ID ranges for specific channels

When using centralized logging in Report Center, to specify the channels from which you want to collect log messages, enter a range of log IDs. You can use individual log IDs to set up triggers. For more information about centralized logging, see the *Symphony Marimba Client Automation Report Center User Guide*. This section lists the ranges for various channels. (An asterisk (*) next to a channel name means that a list of that channel's individual log IDs and messages does not appear in this reference.)

Log ID range	Channel name
1000 - 1999 and 48000 - 51999	“Tuner” on page 427
2000 - 2999	“Transmitter” on page 417
3000 - 3999	Publisher*
4000 - 4999	“Proxy” on page 398
5000 - 5999	“Channel Copier” on page 326
6000 - 6999	“Inventory” on page 371 (Scanner Service and Inventory plug-in)
7000 - 7999	“Tuner Packager” on page 439
8000 - 8999	“Policy Management” on page 387
9000 - 9999	“Application Packager” on page 314
13000 - 13999	“Storage” on page 415
14000 - 14999	Transmitter Guardian*
15000 - 15099 (DM) 15100 - 15399 (DS)	“Deployment Manager and Deployment Service (Server Management)” on page 337
15400 - 18999 (DM)	
19000 - 19999	“Content Replicator (Server Management)” on page 334
24000 - 24999	“Logging” on page 377 (Logging Service and Logging plug-in)
26000 - 26999	Custom Channels*
27000 - 28999	“Common Management Services” on page 328 (CMS)
29000 - 29999	“Report Center” on page 401

Log ID range	Channel name
30000 - 30999	Tuner Update Manager*
31000 - 31999	“Marimba Migration Module” on page 379
32000 - 32999 and 48100 - 51010	“Infrastructure Service” on page 363
34000 - 34999	“Subnet Repeater Policy” on page 416
35000 - 39999	Infrastructure Administration
35000 - 35099	n “Administration tools” on page 313
35100 - 35399	n “Tuner Administrator” on page 435
35400 - 35699	n “Transmitter Administrator” on page 425
35700 - 35999	n “Proxy Administrator” on page 400
36000 - 39999	n “Setup and Deployment” on page 410
40000 - 40999	“Schema Management” on page 408
41000 - 41999	Patch Management
41000 - 41099	n Patch Repository (part of Patch Manager)*
41100 - 41199	n “Patch Service” on page 386
41200 - 41299	n Patch Source (Windows, Linux)*
41500 - 41999	n “Patch Manager” on page 381
42000 - 42999	Configuration Management*
43000 - 43999	Configuration Service*
44000 - 44999	“Action Request” on page 312
50000 - 50999	“Common Reboot Service” on page 333

Action Request

This section lists the log IDs, corresponding log messages, and log severity levels for Action Request.

Log ID	Log message for Action Request	Severity level
44000	Authentication failed	MAJOR
44001	XML parser configuration exception	MAJOR
44002	Failed to initialize call object	MAJOR

Log ID	Log message for Action Request	Severity level
44003	Missing task ID parameter	MAJOR
44004	Incorrect task ID	MAJOR
44005	Web service fault	MAJOR
44006	Missing product dictionary properties	MAJOR
44007	No product dictionary definition for the product	MAJOR
44008	Product dictionary definition is not unique	MAJOR
44009	Missing software package location	MAJOR
44010	No valid SLI instance ID in web services message	MAJOR
44011	Invalid AR settings	MAJOR

Administration tools

This section lists the log IDs, corresponding log messages, and log severity levels for the administration tools in general.

Log ID	Log message for Administration tools	Severity level
35000	Main initialized	AUDIT
35001	Main disposed	AUDIT
35002	Failed to initialize remote-admin manager	MAJOR
35010	Failed to get admin module	MAJOR

For the specific tools, see the following sections:

- “Proxy Administrator” on page 400
- “Transmitter Administrator” on page 425
- “Tuner Administrator” on page 435

Application Packager

This section lists the log IDs, corresponding log messages, and log severity levels for Application Packager.

Log ID	Log message for Application Packager	Severity level
9000	Operation complete	INFO
9001	Operation failed	MAJOR
9002	Preparing to install	INFO
9003	Operation cancelled	WARNING
9004	Verify phase	INFO
9005	Repair phase	INFO
9006	Preload phase	INFO
9007	Download phase	INFO
9008	Backup phase	INFO
9009	Install phase	INFO
9010	Done phase	INFO
9011	Object visited	INFO
9012	Object installed	INFO
9013	Object failed	MAJOR
9014	Object already installed	WARNING
9015	Object verified	INFO
9016	Object attributes modified	WARNING
9017	Object modified	WARNING
9018	Object repaired	INFO
9019	No permissions	MAJOR
9020	Begin Install	INFO
9021	Begin Update	INFO
9022	Begin Uninstall	INFO
9023	Begin Verify	INFO
9024	Begin Repair	INFO

Log ID	Log message for Application Packager	Severity level
9025	Begin Run	INFO
9026	Adapter started	INFO
9027	Adapter cancelled	MAJOR
9028	Exception	MAJOR
9029	Process killed	INFO
9030	Adapter finished	INFO
9031	Script completed	INFO
9032	Abnormal script return	WARNING
9033	Script failure	MAJOR
9034	Critical script failure	CRITICAL
9035	Script execution error	MAJOR
9036	Dependency success	INFO
9037	Dependency failed	MAJOR
9038	Critical dependency failed	CRITICAL
9039	Pre-install script failed	MAJOR
9040	Post-install script failed	MAJOR
9041	Install Failed	CRITICAL
9042	Pre-verify script failed	MAJOR
9043	Post-verify script failed	MAJOR
9044	Verify Failed	MAJOR
9045	Pre-repair script failed	MAJOR
9046	Post-repair script failed	MAJOR
9047	Repair Failed	CRITICAL
9048	Pre-uninstall script failed	MAJOR
9049	Post-uninstall script failed	MAJOR
9050	Uninstall Failed	MAJOR
9051	Pre-launch script failed	MAJOR
9052	Post-launch script failed	MAJOR
9053	Pre-update script failed	MAJOR

Log ID	Log message for Application Packager	Severity level
9054	Post-update script failed	MAJOR
9055	Install succeeded	MAJOR
9056	Adapter info	INFO
9057	Adapter warning	WARNING
9058	Adapter error	MAJOR
9059	Reloaded DLL	WARNING
9060	Rolling Back failed Install	INFO
9061	Begin Rollback after failed install	INFO
9062	Begin Rollback after failed update	INFO
9063	Script failed during Pre-Rollback-Update	MAJOR
9064	Script failed during Post-Rollback-Update	MAJOR
9065	Rolling back after update failed	MAJOR
9066	Script failed during Pre-Rollback-Install	MAJOR
9067	Script failed during Post-Rollback-Install	MAJOR
9068	Rolling back after install failed	MAJOR
9069	Script return code ignored	INFO
9070	Rollback succeed after failed install	MAJOR
9071	Rollback succeed after failed update	MAJOR
9072	Failed to setup MSI log file path	WARNING
9073	Application was already installed so not re-installing	INFO
9074	This channel can only be run on a WTS server	MAJOR
9075	Registering DLL	INFO
9076	UnRegistering DLL	INFO
9077	Cannot copy file	INFO
9078	Cannot create file	INFO
9079	File unwritable, try change permission	INFO
9080	File unwritable, try to recopy after reboot	INFO
9081	File rename failed	INFO

Log ID	Log message for Application Packager	Severity level
9082	Unknown Windows Installer version.	MAJOR
9083	Unsupported Windows Installer version	MAJOR
9084	Windows Installer version	INFO
9085	Windows Installer repair mode	INFO
9086	Removing NT Service on reboot	INFO
9087	File object in channel has no contents	MAJOR
9088	To uninstall a locked file, a reboot is required	INFO
9089	File object in channel has no contents to restore from	INFO
9090	Reinstallation of channel is needed since a repair determined the manifest was obsolete	INFO
9091	Schedule Auto-Repair since content on local filesystem differs for an update containing just file attribute change	INFO
9092	File in filesystem is missing for an update that is modifying attributes	INFO
9093	File in filesystem has different checksum than the one in the manifest	INFO
9094	An operation that is not really valid for the current mode was encountered	INFO
9095	An operation that is not valid was encountered	MAJOR
9096	The adapter is executing in an invalid mode	MAJOR
9097	File object in channel has no contents so it was updated	INFO
9098	Could not obtain write access to the errormode control registry key under Windows	INFO
9099	Successfully registered DLL	INFO
9100	Failed to register DLL	INFO
9101	Successfully unregistered DLL	INFO
9102	Failed to unregister DLL	INFO
9103	DLL is not registrable	INFO
9104	DLL is not unregisterable	INFO

Log ID	Log message for Application Packager	Severity level
9105	Could not create NT service because it was marked for deletion and hasn't been deleted yet	INFO
9106	Failed to create NT service	MAJOR
9107	Failed to install the parent of object	MAJOR
9108	Failed to post install the parent of object	INFO
9109	Failed to read in attributes of object	INFO
9110	Dependent channel required	INFO
9111	Dependent channel does not exist	INFO
9112	Dependent channel subscribed	INFO
9113	Copying DLL from storage to file system during DLL reload	INFO
9114	Not backing up any files	INFO
9115	Removed rollback directory after successful install	INFO
9116	Product not found when doing MSI channel repair	INFO
9117	Product not found on repair, need to redownload file	INFO
9118	Product not found on repair, need to reinstall product	INFO
9119	Deleting file after download, file not mapped	INFO
9120	Operation by MRBAMSIEEXEC	INFO
9121	A reboot is required because it was detected in the registry on Windows NT platforms or in wininit.ini on Windows 9x platforms	INFO
9122	Couldn't open the wininit.ini file to read the rename section	INFO
9123	Object was skipped during verification because verify is not supported for it	INFO
9124	Object failed to be verified	MAJOR
9125	Object failed to be repaired	MAJOR
9126	Script failed during Pre-User-Install	MAJOR
9127	Script failed during Post-User-Install	MAJOR

Log ID	Log message for Application Packager	Severity level
9128	User install failed	MAJOR
9129	Script failed during Pre-User-Update	MAJOR
9130	Script failed during Post-User-Update	MAJOR
9131	User update failed	MAJOR
9132	User install succeeded	MAJOR
9133	User update succeeded	MAJOR
9200	File needs to download contents from a Transmitter for repair	INFO
9201	Object skipped during verification because the policy was never verify	INFO
9202	Object being verified was previously installed by Application Packager	INFO
9203	Object failed to be verified because it does not exist	MAJOR
9204	Object was skipped during verification because it does not exist	INFO
9205	Object failed to be verified because its contents have changed	MAJOR
9206	Object failed to be verified because its contents exist	MAJOR
9207	Object was skipped during verification because its operation does nothing	INFO
9208	Object verification policies	INFO
9209	Object repair policies	INFO
9210	Object was skipped during repair because the policy was never repair	INFO
9211	Object was skipped during repair because the policy was not to repair content	INFO
9212	Object was skipped during repair because its operation does nothing	INFO
9213	No attributes were repaired since the file doesn't exist	INFO
9214	No repair is needed given the current state of the object and the repair policy	INFO

Log ID	Log message for Application Packager	Severity level
9215	Object being verified was not previously installed by Application Packager	INFO
9216	No repair needed for the contents of the file since the checksums match	INFO
9217	Repair needed for the contents of the file since the checksums don't match and the versions have not increased	INFO
9218	Repair needed for the contents of the file since the checksums don't match	INFO
9219	No repair needed for the contents of the file since versions have increased	INFO
9220	Couldn't read the attributes of the file during repair to determine if they match, so setting attributes	WARNING
9221	No repair for the attributes of the file is needed since they already match	INFO
9222	Repair needed for the attributes of the file since they don't match	INFO
9223	Repairing the contents of the file failed	MAJOR
9224	Repairing the attributes of the file failed	MAJOR
9225	No need to unmap file for download from Transmitter because the nofilemap property was true and the file exists in the Tuner storage	INFO
9226	No need to unmap file because no mapping exists	INFO
9227	The adapter will execute in the specified mode	INFO
9228	Object install policies	INFO
9229	Object update policies	INFO
9230	Object uninstall policies	INFO
9231	Object operation to be performed	INFO
9232	Object state before performing operation	INFO
9233	Object state after performing operation	INFO
9234	Couldn't delete the file before restoring from backup	INFO

Log ID	Log message for Application Packager	Severity level
9235	Adapter started, packaged channel URL	INFO
9236	Adapter stopped, packaged channel URL	INFO
9237	Failed to retrieve old manifest from the end point; creating a blank manifest	MAJOR
9238	Failed to parse new manifest from the Tuner storage; please check the original package's manifest.ncp for consistency	MAJOR
9239	Failed to parse old uninstaller on the end point; attempting to create a best effort uninstaller	MAJOR
9240	Best effort recovery yielded a blank uninstaller	MAJOR
9241	Failed to load the object state configuration file, creating blank one	MAJOR
9242	The adapter cannot execute because it has user objects and nobody is logged on to the system	INFO
9243	There was a problem in determining the currently logged on user; defaulting to no one as logged in	MAJOR
9244	Failed to create scripts directory, trying again	INFO
9245	Failed to execute native script, trying again	INFO
9246	Path to native script does not exist	INFO
9247	Failed to launch the native script in a separate process	MAJOR
9248	Failed to set the reference count for the shared file	INFO
9249	Successfully set the reference count for the shared file	INFO
9250	Pre-minorupdate script failed	MAJOR
9251	Post-minorupdate script failed	MAJOR
9252	Performing an update rather than a verify	INFO
9253	Performing an update rather than a repair	INFO
9254	Repairing objects after major update	INFO
9255	Successfully repaired objects after major update	INFO
9256	Failed to repair all objects after major update	INFO
9257	Repairing objects after minor update	INFO

Log ID	Log message for Application Packager	Severity level
9258	Successfully repaired objects after minor update	INFO
9259	Failed to repair all objects after minor update	WARNING
9300	Altering MSI log path because a directory of the same name already exists	INFO
9301	Bypassing MSI APIs and launching commandline	INFO
9302	MSI commandline exit code	INFO
9303	MSI product install type	INFO
9304	Not adding URL to MSI source list in registry because it doesn't support MSI redirection	INFO
9305	Not adding URL to MSI source list in registry because it could not be reached	INFO
9306	First URL from repeater list that supports MSI redirection	INFO
9307	URL is alive and supports MSI redirection	INFO
9308	Setting MSI last used source to	INFO
9309	Adding URL to MSI source list	INFO
9310	MSI redirection Transmitter version	INFO
9311	MSI product code to hashed GUID	INFO
9312	MSI source overridden with	INFO
9313	Staging MSI install	INFO
9314	Staging MSI update	INFO
9315	Cannot stage MSI in this adapter mode	INFO
9316	MSI staging succeeded	INFO
9317	MSI staging failed	MAJOR
9318	MSI channel was staged previously	INFO
9319	MSI product code was previously advertised and will now be installed	INFO
9400	Minor update, check operation	INFO
9401	Minor update, prepare operation	INFO
9402	Minor update, execute pre-scripts	INFO

Log ID	Log message for Application Packager	Severity level
9403	Minor update, execute post-scripts	INFO
9404	Channel files or time file updated	INFO
9405	Channel manifest updated	INFO
9406	Attempted to start a channel that is already running	INFO
9407	There was a mismatch between the channel configuration's edit.time value and the time file value, channel configuration dumped	INFO
9500	Object is already installed, take control	INFO
9501	Object contents match, but wrong attributes. Reset attributes	INFO
9502	Begin Minor Update	INFO
9503	Install Mode succeeded	INFO
9504	Major Update Mode succeeded	INFO
9505	Minor Update Mode succeeded	INFO
9506	Uninstall Mode succeeded	INFO
9507	Verify Mode succeeded	INFO
9508	Repair Mode succeeded	INFO
9509	Run Mode succeeded	INFO
9510	Install Mode failed	INFO
9511	Major Update Mode failed	INFO
9512	Minor Update Mode failed	INFO
9513	Uninstall Mode failed	INFO
9514	Verify Mode failed	INFO
9515	Repair Mode failed	INFO
9516	Run Mode failed	INFO
9517	DLL register/unregister using internal utility	INFO
9518	DLL register/unregister return code	INFO
9519	User Install Mode succeeded	INFO
9520	User Install Mode failed	INFO

Log ID	Log message for Application Packager	Severity level
9521	User Update Mode succeeded	INFO
9522	User Update Mode failed	INFO
9600	Assembly is missing from the Global Assembly Cache	MAJOR
9601	Assembly exists in the Global Assembly Cache when it shouldn't	MAJOR
9602	Assembly exists in the Global Assembly Cache	INFO
9603	Assembly doesn't exist in the Global Assembly Cache	INFO
9604	Attempting to register assembly into Global Assembly Cache	INFO
9605	Successfully registered assembly into Global Assembly Cache	INFO
9606	Failed to register assembly into Global Assembly Cache	MAJOR
9607	Attempting to unregister assembly from Global Assembly Cache	INFO
9608	Successfully unregistered assembly from Global Assembly Cache	INFO
9609	Failed to unregister assembly from Global Assembly Cache	MAJOR
9610	The .NET framework does not exist	INFO
9700	Phase has changed	INFO
9710	Unable to serialize object tree for resume	WARNING
9711	Unable to serialize postprocess objects	WARNING
9712	Need system reboot to resume operation	INFO
9713	Object causing reboot	INFO
9714	Failed to load the object tree to resume the previous operation	MAJOR
9715	Failed to load the object paths to resume the previous operation	MAJOR
9716	No object to do postprocessing after resume	INFO

Log ID	Log message for Application Packager	Severity level
9717	Failed to deserialize object for postprocessing after resume	MAJOR
9718	Failed to perform postprocessing after resume	MAJOR
9719	Successfully postprocessed objects after resume	INFO
9720	Skip postprocessing before resume	INFO
9721	Repair is needed to complete installation	INFO
9722	Invalid resume mode	INFO
9723	Attempted to resume without reboot	INFO
9724	Mode to resume from	INFO
9725	Exit code to resume from	INFO
9726	Script permission ignored due to error; Setting to default: 700	WARNING
9727	Checking local user state	INFO
9728	Current user is not the user who started the operation to resume	MAJOR
9729	Removed local state for current user	INFO
9730	About to update channel	INFO
9731	After updating channel	INFO
9732	Sync failed; unable to update channel	MAJOR
9733	Sync mode failed	INFO
9734	Begin sync	INFO
9735	Begin user install	INFO
9736	Begin user update	INFO
9737	Unable to change adapter mode	INFO

Channel Copier

This section lists the log IDs, corresponding log messages, and log severity levels for Channel Copier.

Log ID	Log message for Channel Copier	Severity level
5001	Copy cancelled	AUDIT
5002	Copy completed	AUDIT
5003	Copy started	AUDIT
5004	Starting segment	AUDIT
5005	Copy cancelled. The following segments were copied	AUDIT
5100	Security unavailable	CRITICAL
5101	Copy exception	CRITICAL
5300	A source exception occurred:	CRITICAL
5301	Source channel has no segments	CRITICAL
5302	Source channel missing index	CRITICAL
5303	Bad command from source transmitter:	CRITICAL
5304	Source transmitter sent unexpected file:	CRITICAL
5305	Source transmitter sent unexpected end of file	CRITICAL
5306	Missing files from source channel:	CRITICAL
5307	Properties file missing from source channel:	CRITICAL
5308	Source transmitter sent bad magic number:	CRITICAL
5309	Source channel not found:	CRITICAL
5310	Source transmitter error:	CRITICAL
5311	Source transmitter protocol not supported	CRITICAL
5312	Source transmitter busy	CRITICAL
5313	Source transmitter sent unexpected reply:	CRITICAL
5314	Bad checksum on source file:	CRITICAL
5315	No additions file	CRITICAL
5316	Invalid additions file entry:	CRITICAL

Log ID	Log message for Channel Copier	Severity level
5317	Addition does not exist:	CRITICAL
5318	Add failed	CRITICAL
5319	Additions exception	CRITICAL
5320	Bad source credentials:	CRITICAL
5321	Source http error	CRITICAL
5322	Source file missing	CRITICAL
5323	Unknown source transmitter	CRITICAL
5324	Could not connect to source transmitter	CRITICAL
5500	A destination exception occurred:	CRITICAL
5501	Destination transmitter is missing files	CRITICAL
5502	Publish to destination transmitter failed	CRITICAL
5503	Unknown destination transmitter	CRITICAL
5504	Could not connect to destination transmitter	CRITICAL
5505	No channel url for CAR file	WARNING
5506	Multiple segments for directory destination	CRITICAL
5507	Bad destination credentials	CRITICAL
5508	Destination http error	CRITICAL
5509	Host is not authorized to publish to destination	CRITICAL
5510	No SSL port available on destination	CRITICAL
5511	Cannot write to destination	CRITICAL
5512	Publish forbidden on destination	CRITICAL
5513	Publish service unavailable on destination	CRITICAL
5600	Could not find channel signing certificate:	CRITICAL
5601	Channel signing failed	CRITICAL
5602	Operation will break channel signature. Edited properties	CRITICAL
5603	Warning: certificate is expired	WARNING
5700	Starting to delete	AUDIT
5701	Starting to delete segment	AUDIT

Log ID	Log message for Channel Copier	Severity level
5702	Finished deleting segment	AUDIT
5703	Finished deleting	AUDIT
5750	Delete failed	CRITICAL
5751	Failed to delete segment	CRITICAL
5752	Error in delete. Destination not a transmitter	CRITICAL
5753	Segment does not exist on destination	WARNING

Common Management Services

This section lists the log IDs, corresponding log messages, and log severity levels for Common Management Services (CMS).

Log ID	Log message for CMS	Severity level
27000	ServletContainer initialized	AUDIT
27001	ServletContainer disposed	AUDIT
27002	Application added	AUDIT
27003	Application removed	AUDIT
27004	Error adding application	CRITICAL
27005	Error removing application	CRITICAL
27006	User logon	AUDIT
27007	User logout	AUDIT
27008	Failed logon attempt	AUDIT
27010	Failed to create server	CRITICAL
27011	Failed to restart server	CRITICAL
27012	Server restarting	AUDIT
27013	Another instance is already running	MAJOR
27014	Java 2 VM required, current VM version	CRITICAL
27015	Failed to bind to port	MAJOR
27016	Missing application with context-path	MINOR
27017	Timeout waiting for application to launch	MINOR

Log ID	Log message for CMS	Severity level
27020	Changed server root-directory	AUDIT
27021	Changed server log-directory	AUDIT
27030	LDAP connection established	AUDIT
27031	Failed to initialize LDAP - invalid configuration	MAJOR
27032	Failed to connect to LDAP server	CRITICAL
27033	LDAP server connection lost	CRITICAL
27050	Application Audit	AUDIT
27051	Application Error	MINOR
27060	SSL enabled using certificate	AUDIT
27061	SSL disabled	AUDIT
27062	SSL error	CRITICAL
27070	Running in standalone mode	AUDIT
27071	Running in master mode	AUDIT
27072	Running in slave mode	AUDIT
27073	Global configuration published	AUDIT
27074	Error publishing global configuration	MINOR
27075	Updating global configuration	AUDIT
27076	Error updating global configuration	MINOR
27100	Failed to get real-path of URI	MINOR
27110	Failed to load servlet on startup	MAJOR
27150	Request error	MAJOR
27151	Failed to send error reply	MAJOR
27152	Failed to parse POST-data	MAJOR
27153	Unexpected error	MAJOR
27154	Request statistics	AUDIT
27155	Request closed by client	MINOR
27200	Error reply failed	MAJOR
27201	Failed to send redirect	MAJOR
27250	Authenticator initialized	CRITICAL

Log ID	Log message for CMS	Severity level
27251	Failed to dispose authenticator	MAJOR
27252	Lookup failed	MAJOR
27300	Received command-line arguments	AUDIT
27350	Connection established	AUDIT
27351	Connection-pool disconnected	CRITICAL
27352	Connection-pool driver	AUDIT
27353	Connection-pool error	CRITICAL
27354	Connection-pool enabled	AUDIT
27355	Connection-pool disabled	AUDIT
27400	Applications Manager start	AUDIT
27401	Applications Manager stop	AUDIT
27402	Applications Manager update	AUDIT
27403	Applications Manager remove	AUDIT
27404	Applications Manager subscribe	AUDIT
27410	Authentication settings changed	AUDIT
27411	Failed to change authentication settings	WARNING
27415	E-mail server settings changed	AUDIT
27420	LDAP configuration changed	AUDIT
27421	LDAP configuration added	AUDIT
27422	LDAP configuration deleted	AUDIT
27430	Local User Database - user added	AUDIT
27431	Local User Database - user deleted	AUDIT
27432	Local User Database - user edited	AUDIT
27440	LDAP RoleMappings changed	AUDIT
27450	Emergency Administrator password changed	AUDIT
27460	DatabaseManager - database added	AUDIT
27461	DatabaseManager - database deleted	AUDIT
27462	DatabaseManager - database edited	AUDIT
27470	Port changed	AUDIT

Log ID	Log message for CMS	Severity level
27471	Bind-address changed	AUDIT
27480	User-timeout changed	AUDIT
27481	Restart server	AUDIT
27490	Log Settings changed	AUDIT
27500	Task added	AUDIT
27501	Task deleted	AUDIT
27502	Task loaded	AUDIT
27503	Task unloaded	AUDIT
27504	Task initialized	AUDIT
27505	Task executing	AUDIT
27506	Task done	AUDIT
27507	Task failed to execute	MAJOR
27508	Host application is not running for task	WARNING
27509	Task has an invalid schedule	MINOR
27510	Task has an invalid trigger	MINOR
27511	An error occurred while evaluating the trigger for	MINOR
27512	Trigger expression return non-boolean value	MINOR
27513	Error initializing task	MAJOR
27514	Error destroying task	MAJOR
27515	Bad pipe missing receiving task	WARNING
27516	Task triggered	AUDIT
27550	Storage error	CRITICAL
27551	Attempt to store invalid task	MAJOR
27552	Bypassing throttling mechanism	WARNING
27600	ACL add called	AUDIT
27601	ACL add succeeded	AUDIT
27602	ACL add failed	MAJOR
27603	ACL delete called	AUDIT
27604	ACL delete succeeded	AUDIT

Log ID	Log message for CMS	Severity level
27605	ACL delete failed	MAJOR
27606	ACL initialization succeeded	AUDIT
27607	ACL initialization failed	MAJOR
27700	Successfully sent email	AUDIT
27701	Failed to connect to the SMTP server	MAJOR
27702	Invalid email configurations	MAJOR
27703	Error occurred while sending email	MAJOR
27750	Unable to create System directory under tuner's installation directory	MAJOR
27751	Error occurred while copying CMDB native library file	MAJOR
27752	Missing CMDB native library directory	MAJOR
47201	No active database configured in CMS	LOG_MAJOR
47202	Database connection cannot be established	LOG_MAJOR
47203	Error in loading seed tuner	LOG_MAJOR
47204	No seed tuner found	LOG_AUDIT
47205	Failed to wakeup machines	LOG_AUDIT
47206	Failed on loading machines	LOG_AUDIT
47207	Failed to logon	LOG_INFO
47208	Invalid url	LOG_INFO
47209	PSS loaded in CMS	LOG_INFO
47210	PSS loading targets	LOG_INFO
47211	Threads Info	LOG_INFO
47212	Starting Subnet	LOG_INFO
47213	Subnet Wake Info	LOG_INFO
47214	SeedFinder Info	LOG_INFO
47215	LMS Info	LOG_INFO
47216	vPro Info	LOG_INFO
47217	Wake Admin	LOG_INFO

Log ID	Log message for CMS	Severity level
47218	DB Info	LOG_INFO
47219	DB Error	LOG_MAJOR
47220	LDAP Info	LOG_INFO
47221	LDAP Error	LOG_MAJOR
47222	Subnet Queued	LOG_AUDIT
47223	Subnet Running	LOG_AUDIT
47224	Subnet Succeeded	LOG_AUDIT
47225	Subnet Failed	LOG_AUDIT
47226	Subnet Cancelled	LOG_AUDIT
47227	Cache Info	LOG_INFO
47228	Failed to parse subnet filter	LOG_INFO
47229	Initialized WoW context	LOG_INFO
47230	WoW execution strategy order	LOG_INFO
47231	Database connection dropped	LOG_MAJOR
47232	Database connection established	LOG_MAJOR
47233	Subnet Info	LOG_INFO

Common Reboot Service

This section lists the log IDs, corresponding log messages, and log severity levels for the Common Reboot Service.

Log ID	Log message for Administration tools	Severity level
50001	Invalid Reboot Priority - Ignoring System Reboot.	LOG WARNING
50002	Service channels are running - Delaying System Reboot!	LOG INFO
50003	CRS Triggered System Reboot at :- <current Time>	LOG INFO
50004	Reboot is scheduled - Scheduled Reboot Time < next Schedule Time>	LOG INFO
50005	Tuner processing pending system reboot	LOG INFO

Log ID	Log message for Administration tools	Severity level
50006	Failed to parse last reboot time	LOG WARNING
50007	Invalid reboot priority: <priority>. Defaulting to Normal (1) priority.	LOG WARNING
50008	Stored boot time: <last known system boot Date time>, Retrieved boot time: <new system boot Date time>	LOG INFO
50009	Failed to retrieve system reboot time.	LOG WARNING
50010	Channel got abruptly stopped. Clearing reboot request that was requested by channel: <channel Url>	LOG WARNING
50011	IO Exception	LOG WARNING
50012	Common Reboot Service is disabled.	LOG WARNING
50013	Reboot already in progress. Reboot request by channel: <channelUrl> ignored.	LOG WARNING
50014	Error occurred while reading resource file : <file name>	LOG MAJOR

Content Replicator (Server Management)

This section lists the log IDs and corresponding log messages for Content Replicator.

Log ID	Log message for Content Replicator	Severity level
19000	Info	minor or info
19001	Success	major
19002	Error	major
19003	Warning	major or minor or info or warning
19010	Publishing	warning or info
19011	Publish completed successfully	info
19012	Publish warning	minor or warning

Log ID	Log message for Content Replicator	Severity level
19013	Publish error	major or minor
19020	Deleting	info
19021	Delete completed successfully	info
19022	Delete warning	warning or minor
19023	Delete error	major or minor
19030	Installing	info or warning
19031	Install completed successfully	info
19032	Install warning	minor or warning
19033	Install error	warning or major or minor
19040	Rolling back	info or warning
19041	Rollback completed successfully	info
19042	Rollback warning	minor or warning
19043	Rollback error	warning or major or minor
19050	Hinting	info
19051	Hint completed successfully	info
19052	Hint warning	minor or warning
19053	Hint error	warning
19060	Previewing	info
19061	Preview completed successfully	info
19062	Preview warning	warning
19063	Preview error	warning
19070	Uninstalling	info
19071	Uninstall completed successfully	info
19072	Uninstall warning	warning
19073	Uninstall error	major

Log ID	Log message for Content Replicator	Severity level
19080	Progress	info
19081	Progress	info
19082	Progress	info
19090	Prefs	info
19100	Validating	info
19101	Validate completed successfully	info
19102	Validate warning	info
19103	Validate error	major or minor
19104	Verifying	warning
19105	Verifying	warning
19106	Verifying	warning
19107	Verifying	warning
19108	Verifying	warning
19109	Verifying warning	warning
19200	Installing Unix package	info
19201	Unix package installation completed successfully	info
19202	Unix package installation warning	info
19203	Unix package installation error	critical or major
19210	Uninstalling Unix package	info
19211	Unix package un-installation completed successfully	info
19212	Unix package un-installation warning	info
19213	Unix package un-installation error	critical or major
19220	Verifying Unix package	info
19221	Unix package is verified to be valid	info
19222	Unix package verification warning	info
19223	Unix package verification error	major or critical
19224	Unix package is not valid	major
19230	Repairing Unix package	info

Log ID	Log message for Content Replicator	Severity level
19231	Unix package repair completed successfully	info
19232	Unix package repair warning	info
19233	Unix package repair error	critical or major
19240	Updating Unix package	info
19241	Unix package update completed successfully	info
19242	Unix package update warning	info
19243	Unix package update error	major or critical

Deployment Manager and Deployment Service (Server Management)

This section lists the log IDs and corresponding log messages for Deployment Manager and Deployment Service.

Note: A caret (^) followed by a number indicates that the name of a specific object, such as a server or server group, is inserted in that location of the log message. For example, in the log message “An error occurred while adding server ‘^0’ to server group ‘^1.’” ^0 is replaced by a server name and ^1 is replaced by a server group name.

Log ID	Log message for Server Management
15000	Credentials changed: ^0
15001	Server restart
15002	Server starting
15003	Unable to load the tree. Please check the Deployment Manager logs for a stack trace.
15004	Deployment ‘^0’ value for ^1 will be lost because this property now applies at the job level, and there are no jobs.
15005	An error occurred while adding server ‘^0’ to server group ‘^1’.
15006	An error occurred while loading object in directory ‘^0’.
15007	An error occurred while importing an object of type ‘^0’.

Log ID	Log message for Server Management
15008	Invalid storage version number '^0', assuming version 1.x storage.
15009	Created temporary directory '^0' for use during ^1.
15010	The SNMP traps will not be sent for '^0' because the SNMP community list, ^1, cannot be used with the SNMP station list, ^2. The community list must either be one value or contain the same number of communities as stations.
15011	The SNMP traps cannot be sent to the SNMP server, ^1, for '^0'. This is because the SNMP server host name specified cannot be resolved by the machine on which Deployment Manager is running.
15012	Deployment Manager stopped.
15014	Command-line invocation failed. Arguments: ^0
15015	License not found for this product.
15016	Failed to send SNMP traps for '^0' because: ^1
15017	Successfully sent SNMP trap ('^2') for '^0' to host name '^1'.
15050	Command to Deployment Manager succeeded.
15051	Command to Deployment Manager failed. ^0
15052	SOAP error: too many elements while trying to add element '^0'.
15053	SOAP error: not enough elements while trying to add element '^0'.
15054	SOAP error: no more element specs while trying to add element '^0'.
15055	SOAP warning: ignoring data for current element '^0'.
15060	File unzipped successfully.
15061	XML file parsed and DOM created.
15215	Client status
15216	Error creating log
15217	Waiting for log entries
15218	Channel request error
15219	Channel request access error
15220	Channel request canceled
15221	Illegal command: ^0
15222	Illegal state: ^0

Log ID	Log message for Server Management
15223	Channel subscribe error: ^0
15224	Channel command error: The channel command, ^0, failed because the channel ^1 ^2.
15225	Channel start error: ^0
15226	Job starting
15227	Starting command: ^0
15228	Command failed: ^0
15229	Command succeeded: ^0
15230	Job failed on endpoint
15231	Job succeeded on endpoint
15232	Job stopped on endpoint
15233	The endpoint is able to communicate with the Deployment Manager. Status URL: ^0
15234	Job restart on endpoint: ^0
15235	Deployment load error: ^0
15236	Deployment Service error: The endpoint ^0 failed due to an error while attempting to write to the disk. Check to make sure that there is enough disk space available on the endpoint machine.
15237	Client exited unexpectedly.
15238	Remote system command error: The remote system command, ^0, has been terminated by the Deployment Service because the command response timeout of, ^1, expired before the command completed. A larger value for the command response timeout might prevent this from occurring.
15239	Remote system command error: The remote system command, ^0, was killed by another process before it completed.
15240	Channel command error: The channel command has exited, yet it is continuing to generate log entries. Please contact the distributor of the software to report this error.
15241	Channel command error: The channel command has exited before generating a success or failure status log entry.
15242	Server processing...
15243	Starting command: ^0 (macro-expanded: ^1)

Log ID	Log message for Server Management
15245	Unable to roll back because no backup was saved previously. The channel did not have the undo property (undo.count) specified to at least 2.
15246	The undo.count property was set to a non-integer. Cannot perform rollback.
15247	The undo.current was set incorrectly by the system. Cannot roll back.
15248	Attempting to roll back more than is possible. No more previous indices.
15249	Failed while attempting to revert to the previous version.
15251	System Command returned exit status: ^0
15252	Only one deployment job of a given type is allowed to run on an endpoint at a time.
15259	Channel update error: ^0
15272	Client output: ^0
15273	Unable to start the job on the server endpoint. This happens if an error has occurred in Deployment Service. The summary error message that occurred is '^0'. Please contact the distributor of this software to report this error. Be prepared to provide the logs of the Deployment Service channel on the server endpoint.
15274	Unable to start the job on the server endpoint. This happens when the URL sent to the Deployment Service from Deployment Manager is not correctly formed. This error should not occur. Please contact the distributor of the software to report this error. The URL sent to the Deployment Service was '^0'.
15275	Unable to start the job on the server endpoint. This happens when the URL sent to the Deployment Service from Deployment Manager is incorrect. This error should not occur. Please contact the distributor of the software to report this error. The URL sent to the Deployment Service was '^0'.

Log ID	Log message for Server Management
15276	Unable to start the job on the server endpoint. This occurs when the server endpoint is unable to resolve the host name of the machine hosting Deployment Manager. Currently, the Deployment Manager host name value is ^0. This could occur if the DNS settings on the server endpoint cannot resolve the Deployment Manager host name or if a proxy between the Deployment Service and Deployment Manager does not allow requests. Status is sent back to Deployment Manager using the host name that is set in the Settings page. If you do not have access to change the host name, contact your Deployment Manager administrator.
15277	Unable to start the job on the server endpoint. Deployment Manager has refused status from the server endpoint. This could occur if Deployment Manager is too busy to accommodate more status messages. Please decrease your active server limit for the job. Contact the distributor of this software for other ways to increase the number of status posts Deployment Manager can handle. Another possibility is that the endpoint cannot resolve the DNS name of the machine hosting Deployment Manager. To remedy this situation, you will need to use the machine's IP address instead of its host name in Deployment Manager's System Settings.
15278	Unable to start the job on the server endpoint. The server endpoint cannot send status to Deployment Manager because no network path to Deployment Manager could be found. This might occur if there is an intervening firewall or if an intermediate router is down. Please connect to the server endpoint and make sure that the URL, ^0, can be reached.
15279	Unable to start the job on the server endpoint. Communication cannot be established from the server endpoint to Deployment Manager. The network may have gone down during the connection. The Deployment Manager URL being used is ^0.
15280	Unable to start the job on the server endpoint. The SSL certificate used on the machine hosting Deployment Manager is not valid for use by this server endpoint. The Deployment Manager URL being used by the Deployment Manager is ^0. The root certificate must be trusted for SSL use in the certificate database of the Tuner on the server endpoint.
15281	Unable to start the job on the server endpoint. The root certificate of the SSL certificate used by the machine hosting Deployment Manager is not valid for use by this server endpoint. The Deployment Manager URL being used is ^0. The root certificate must be trusted for SSL use in the certificate database of the Tuner on the server endpoint.

Log ID	Log message for Server Management
15282	Unable to start the job on the server endpoint. The Deployment Manager host name used by the server endpoint does not match that of the common name of Deployment Manager's SSL certificate. Ensure that the host name in the Settings page of Deployment Manager matches that of the common name of the SSL certificate used. The current host name is ^0.
15283	Unable to start the job on the server endpoint. There was an error in the network communication between Deployment Manager and the Deployment Service. This could occur if network connection was lost. The Deployment Manager URL used is ^0. The error code that resulted was ^1.
15284	Unable to start the job on the server endpoint. This occurs if the information that a proxy uses for verification when sending status reports to Deployment Manager is invalid. This should not occur. Please contact the distributor of your proxy to ensure that it can handle SSL requests.
15285	Unable to start the job on the server endpoint. This occurs if the information that a proxy uses for verification when sending status reports to Deployment Manager is incorrect. This should not occur. Please contact the distributor of your proxy to ensure that it can handle SSL requests.
15286	Unable to start the job on the server endpoint. This occurs if the information that a proxy uses for verification when sending status reports to Deployment Manager is incorrect for establishing an SSL connection. Please ensure that your proxy can handle SSL requests.
15287	Unable to parse command: ^0. This can happen due to an error in the Deployment Manager allowing an invalid command format to be saved or corruption of data during transmission. Check the command format, and retry the job. If the problem persists report this error.
15288	Channel not found: ^0
15289	UpdateFrom URL missing: ^0
15290	Channel update failed: ^0
15291	Client output: ^0
15292	Client error: ^0
15293	The URL ^1 in the channel command ^0 contains a space which is not allowed for channel URLs.

Log ID	Log message for Server Management
15294	Waiting to run channel ^0. Already max instances (^1) of the channel are running. Queue position: ^2.
15295	Running remote system command as the user: ^0
15296	Running SDMClient Version: ^0
15297	Could not resolve value of the macro: ^0
15350	Unknown Deployment Manager, ^0. This indicates that the network connection was lost while the commands were running for ^1. Verify that Deployment Manager server is up and still accessible from this endpoint. The connection will be retried.
15351	A network route to Deployment Manager, ^0, could not be found. This indicates that the network connection was lost while the commands were running for ^1. The connection will be retried.
15352	The network connection was not handled by Deployment Manager, ^0. This occurs if the server is too busy to handle status. In Deployment Manager, please decrease your active server limit for the job. Contact the distributor of this software for other ways to increase the number of status posts Deployment Manager can handle. The connection will be retried.
15353	The network connection was not handled by Deployment Manager, ^0. This occurs if the server rejects a seemingly valid status post. This can occur if Deployment Manager is too busy with other status post. The connection will be retried.
15354	Failed to send status for ^0. This could occur if Deployment Manager no longer is listening for status about this job.
15355	The network connection was not handled by Deployment Manager, ^0. The connection will be retried.
15356	Failed to send status for ^0. The network connection to Deployment Manager will be retried.
15357	Failed to send status for ^0. The error that occurred was ^1. The connection will be retried.
15358	The maximum number of retries was reached for ^0. A network connection could not be established for sending back status.
15359	Failed to send status for ^0. The network connection to Deployment Manager will be retried.
15360	An error occurred while running a command. Internal error in Command thread. Exception: ^0.

Log ID	Log message for Server Management
15361	An error occurred while running a command. Internal error in Active thread. Exception: ^0.
15362	Reacquiring file lock: ^0
15363	Reacquiring file lock failed: ^0
15364	Reacquiring file lock done: ^0
15365	Timeout occurred trying to acquire lock.
15366	Couldn't access or create lock directory.
15367	Encountered a stale lock
15368	Failed to create temp file in lock directory. Check write permissions or space in lock directory.
15369	Encountered unknown error attempting to acquire lock
15370	Unlock failed because lock not owned by current process
15371	Unlock failed because attempting to remove lock file failed
15373	Do not have permissions to create file in the lock directory: ^0
15374	Failed to store creation of receipt in SDMClient: ^0
15375	Failed to upload logs
15376	Upload logs ^0
15377	Initial request ^0
15378	Sending heartbeat request
15379	Sending getfile request ^0
15380	Flushing log queue ^0
15400	Job stopped on server: ^0. Stopped by: ^1. Reason: ^2.
15401	The server credentials used do not work for connecting to the server endpoint, '^0'. The credentials used were obtained from the server keychain, '^1'.
15402	Job failed.
15403	Job stopped. Stopped by: ^0. Reason: ^1.
15404	Job started even though its parent deployment is not part of the folder hierarchy. This should not occur. Please contact your distributor of this software to file this problem: ^0
15405	Job succeeded.

Log ID	Log message for Server Management
15406	Illegal start schedule: ^0
15407	Connection to server failed: ^0
15408	Unable to subscribe to the Deployment Service channel,^0, on the server endpoint,^1. The deployment Service channel URL was derived from the ^2. This error can occur if the server endpoint is unable to subscribe to the specified URL.
15409	Deployment Manager has not received a response from server endpoint for ^0 minutes, the value specified for the server response timeout. Therefore, Deployment Manager is failing the job on this server endpoint. An endpoint can time out if the machine has gone down, if the network communication is slow at the endpoint, or if Deployment Manager is too busy to handle the endpoint status right away. Try increasing the server response timeout and/or the grace period.
15410	Command timeout.
15411	Cannot start the job because there are no server groups upon which to act.
15412	Cannot start the job because there is no task group associated with the deployment. This should not occur. Please contact the distributor of this software to report this error.
15413	Cannot run the deployment with incomplete mail info: ^0.
15414	Status from SMTP mailer: ^0.
15415	Deployment Manager cannot resolve the server URL, '^0'. Check the DNS server settings on the machine that hosts Deployment Manager to ensure that the server URL can be resolved.
15416	The server endpoint cannot be reached because either the machine is down or there is something preventing communication with the server endpoint. Possibilities include a firewall, invalid proxy settings on the machine hosting Deployment Manager, or an intermediate router is down.
15417	Cannot connect to the server endpoint. Either the protocol specified ('^0'), port('^1'), or the server URL('^2') is incorrect for connecting to the server endpoint.
15418	Unable to run the job on the server endpoint. An error has occurred in the Deployment Service on the server endpoint. Please check the Deployment Service channel logs on the server endpoint. This error should not occur. Please contact your software distributor.

Log ID	Log message for Server Management
15419	Unable to start the Deployment Service on server endpoint, '^0'. An error has occurred on the server endpoint. Check the logs for the Deployment Service channel on the endpoint.
15420	A job was running even though its parent deployment was not part of the folder hierarchy. '^0.^1. This is an internal error and should not occur. Please contact your software distributor.
15421	The job is already running: '^0'
15422	Server group failed because the minimum number of servers that had to succeed (quorum) was not achieved. Server group: ^0. Number of failed servers: ^1. Total servers in server group: ^2.
15423	The server group, ^0, succeeded. Server status: ^1 succeeded and ^2 failed.
15424	Starting the server group, ^0. Its run order is ^1.
15425	Unable to start the job, ^0, due to internal errors. This should not occur. Please contact the distributor of this software to report this error. Internal Exception: ^1
15426	Unable to initialize the job, ^0, due to internal errors. This should not occur. Please contact the distributor of this software to report this error. Internal Exception: ^1
15427	Job started by the user, ^0.
15428	Scheduled start of job. The job is using the user name ^0 for all permission verification. This is the user who scheduled the job.
15429	Job started as a '^0' job for ^1. The job that initiated the start of this job can be found at ^2.
15430	Unable to start job, ^0, because the user, ^1, does not have permission to start this job. Execute permission is required on the job's parent deployment.
15431	Job, ^0, failed because verification that is done before a job runs failed.
15432	The job, ^0, could not be started because the job properties for the run could not be initialized. Reason: ^1.
15433	The job could not be started because initialization of some of the properties failed. This should not occur. Please contact the distributor of this software to report this error. Internal error: ^0
15434	The job specified for '^0' was invalid. This should not occur. Please contact the distributor of this software to report this error.

Log ID	Log message for Server Management
15435	The job specified for '^0' does not exist. This should not occur. Please contact the distributor of this software to report this error.
15436	The job specified for '^0' has been deleted. All jobs referenced in the starting job properties must exist in the system in order for the job to start. Value specified for '^0' is '^1'.
15437	The job specified for '^0' does not exist. All jobs referenced in the starting job properties must exist in the system in order for the job to start. Value specified for '^0' is '^1'.
15438	The job, '^0', specified for '^1' cannot be started since the user, '^2', does not have execute permission. In order to start a job, the user must have execute permissions for all jobs referred to in the job properties.
15439	The job cannot be started because the value for '^0' cannot be added to the job properties. This should not occur. Please contact the distributor of this software to report this error.
15440	The job cannot start because the extended type cannot verify that it can start. This should not occur. Please contact the distributor of this software to report this error. Internal error to report: ^0
15441	The job cannot be started because it refers to a server group that has been deleted from the system. Server group, '^0', at index, '^1', is the server group referred to, which has been deleted.
15442	The job cannot be started because the server group referenced at index, ^0, of the server group list does not exist in the system. This should not occur. Please contact the distributor of this software to report this error.
15443	The job cannot be started because the user, '^0', does not have execute permission. In order to start a job, the user must have execute permissions on all of the server groups referred to in the deployment server group list.
15444	The job cannot be started because the server group, ^0, cannot be obtained. This should not occur. Please contact the distributor of this software to report this error.
15445	The job cannot be started because there are no servers in the server group, ^0, at run order index ^1. Each server group must have at least one server in order for the job to start.
15446	The job cannot start. Unable to obtain the list of server groups on which to act. This should not occur. Please contact the distributor of this software to report this error.

Log ID	Log message for Server Management
15447	The job cannot start because the servers for the dynamic server group, '^0', cannot be loaded into the system.
15448	The job cannot start. Unable to obtain a task group. This should not occur. Please contact the distributor of this software to report this error.
15449	Unable to obtain the commands from the task due to an internal error. Internal error to report: Unable to locate the task group in the system due to problems when loading. The identifier for the task group is ^0.
15450	The job cannot be started because it refers to a task group that has been deleted from the system. The task group referred to is ^0.
15451	The job cannot be started because the task group, ^0, cannot be obtained. Please view the task group to see if there are problems. This should not occur.
15452	The job cannot be started because the task, '^0', cannot be obtained. This should not occur. Please contact the distributor of this software to report this error.
15453	The job cannot be started because the corresponding task, '^0', does not contain any commands.
15454	The job cannot be started. No credentials were found for the server, '^0'. The server belongs to server group, '^1'.
15455	The server could not start because the server URL specified, '^0', is invalid. This should not occur because verification was done before the server was added to the server group. Please contact the distributor of this software to report this error.
15456	Unable to get all of the commands needed to run on the servers. Error Message: '^0'
15457	Unable to obtain all of the information needed to run the servers in server group, ^0, at location, '^1' in the deployment server group list. Error Message: '^2'
15458	The pre-job failed to start for server group, '^0', which is located at line, '^1', of the list of server groups of the deployment.
15459	The job cannot start because the macros that the integration module for the deployment adds before job runs cannot be added. This should not occur. Please contact the distributor of this software with this problem. The macros from the extended type were '^0'. The exception thrown was '^1'.

Log ID	Log message for Server Management
15460	The job cannot start because the macros sent to this job from '^0' cannot be added to the list of commands to run. This should not occur. Please contact the distributor of this software with this problem. The macros were '^0'. The exception thrown was '^1'.
15461	The job on server, '^0', cannot be started because the commands to run cannot be initialized. This should not occur. Please contact the distributor of this software to report this error.
15462	The job on server, '^0', cannot be started because the server group macros cannot be added to the list of commands to run. This should not occur. Please contact the distributor of this software to report this error.
15463	The job on server, '^0', cannot be started because the server macros cannot be added to the list of commands to run. This should not occur. Please contact the distributor of this software to report this error.
15464	The job is starting server group, '^0', which is located at line '^1' in the list of server groups of the deployment.
15465	The job cannot be started because the user, '^0', does not have execute permission. In order to start a job, the user must have execute permissions on the task group referred to in the deployment to which this job belongs.
15466	The job specified for '^0' cannot be found. This should not occur. Please contact the distributor of this software to report this error.
15467	The job specified for '^0' has been deleted from Deployment Manager. The job specified was '^1'.
15468	The job cannot be started because the value specified for '^0' is invalid. Check the job run properties to determine where this value was obtained.
15469	The job specified for '^0' cannot be validated; therefore, it cannot start.
15470	Job failed because the job specified for '^0' cannot be validated before starting.
15471	Job failed because the job specified for '^0' could not be successfully started. The job specified was '^1'. The error that occurred that prevented it from starting was '^2'.

Log ID	Log message for Server Management
15472	The job specified for '^0' could not be successfully started. The job specified was '^1'. The error that occurred that prevented it from starting was '^2'.
15473	The job specified for '^0' was successfully started. The job specified was '^1'. The job run page can be viewed at ^2.
15474	Unable to connect to server endpoint, '^0'. An error occurred that is preventing the connection, '^1'.
15475	Unable to connect to server endpoint because there is no Tuner listening for connection requests on the port specified in the server endpoint URL, '^0'.
15476	Job failed because a job that can be specified to run before and after server groups in a job run cannot itself have a job that runs before and after server groups. That is, a job that is a pre-job or post-job cannot itself have a pre-job or post-job associated with it.
15477	Job failed because the job specified for '^0' cannot be validated before starting.
15478	Job failed because the job specified for '^0' could not be successfully started. The job specified was '^1'. The error that occurred that prevented it from starting was '^2'.
15479	Job run failed because the job run, '^1' specified for '^0' did not succeed. The job run page can be viewed at ^2.
15480	The job, '^1', specified for '^0' succeeded. The job run page can be viewed at ^2.
15481	Unable to get information on the most recent job run due to an internal error. Internal error to report: Can't get '^0' property from quick status for job '^1'.
15482	Unable to obtain the commands from the task due to an internal error. Internal error to report: The task group and task are not in Deployment Manager. This is due to a problem with loading the task group. The identifier for the task group is ^0.
15483	Attempting to connect to server, '^0', in order to send the job to run.
15484	This server endpoint has already run in the server group, ^0, and will not run again. For purposes of obtaining an accurate value for '^1', the reported status for this server endpoint will be '^2'. This was obtained from first server endpoint with this URL to run for the job.

Log ID	Log message for Server Management
15485	This server endpoint has already run for this job run. However, the status information for the first time this server endpoint ran cannot be obtained. Therefore, the status of '^0' will be assumed for obtaining an accurate value for '^1'.
15486	The job cannot start because the macros that the extended type for the deployment adds after job runs cannot be added. This should not occur. Please contact the distributor of this software to report this error. The macros from the extended type were '^0'. The exception message was '^1'.
15487	The job cannot start because the task group macros cannot be added. This should not occur. Please contact the distributor of this software to report this problem. The macros from the extended type were '^0'. The exception thrown was '^1'.
15488	The job is not running: '^0'
15489	Unable to start the job because the user, ^0, does not have permission to start this job. The user who schedules a job to run must have execute permissions on the job's parent deployment.
15490	Warning: log entry has invalid format: ^0
15491	Unable to find status for job specified for '^0'. The job run name is ^1, job run ID ^2. This could occur if the system has run out of disk space. Failure will be assumed.
15492	Unable to find status for job specified for '^0'. The job run name is ^1, job run ID ^2. This occurs if the status value has been corrupted. Assuming failure for the '^0'.
15493	The post-job failed to start for the server group, '^0', which is located at line, '^1', of the list of server groups of the deployment.
15494	Starting ^0, ^1.
15495	Job succeeded with errors. Check the job run log entries for details with - minor - severity.
15496	The pre-job, ^0, is already running. This job run will wait until the pre-job is finished and then run the pre-job for this job run.
15497	The post-job, ^0, is already running. This job run will wait until the post-job is finished and then run post-job for this job run.
15498	The job specified for '^0' can no longer be used for a pre-job or a post-job. This occurs if the setting, ^1, on the job specified, ^2, has been changed to 'no'.

Log ID	Log message for Server Management
15499	An error occurred while processing command status for the server, ^0. This is not a major error, but should be reported to the distributor of this software. The status was ^1. The status ID was ^2.
15500	An error occurred while processing the status for the server, ^0. This is not a major error, but should be reported to the distributor of this software. The status was ^1.
15501	The job cannot run on the server, ^0, because the user (^2) starting the job does not have write permissions on ^1. When a Deployment Service changePassword command is part of the task, the user must have write permissions on the keychain from which the servers obtain their credentials.
15502	Deployment Manager could not complete the connection to the endpoint within the server response timeout period. Please verify that the endpoint can still receive connection requests. This also could be caused by intermittent network errors or slow network connections. Please increase the server response timeout and try running the job again.
15503	The current job ^0 (active ID ^1) is no longer running. Deployment Manager is clearing this inactive job run to a new job run to start.
15504	Unable to create an aggregate server group from the server group list ^0 of the deployment '^1'. Error Message: '^2'
15505	Unable to start job, ^0, because the user, ^1, has been disabled from the directory server.
15506	Not allowed to stop pre/post job '^0'. You may not manually stop pre/post jobs. If you would like to stop this job, you must stop the original job that initiated it.
15507	Not allowed to start pre/post job '^0'. You may not manually start pre/post jobs. If you would like to start this job, you must start the original job that will initiate it.
15508	Error: No server view log filename specified.
15509	Invalid server viewing filename specified.
15510	Server run not found
15511	Dmp module request
15515	Failed to create retry job
15516	Failed to run retry server group '^0' in job '^1' due to '^2'.

Log ID	Log message for Server Management
15517	System retrying server group, '^0', which is located at line '^1' of server groups of the deployment.
15518	Failed to retry server '^0' from job '^1' because the server group '^2' it belongs to no longer exists.
15519	Failed to retry server '^0' from job '^1' because the server keychain '^2' for it no longer exists.
15520	Retry of job '^0' is not needed because there are no failed servers.
15521	Can not retry job '^0' because it has never ran before. If you want to run this job, start it normally.
15522	Can not get credentials from server keychain '^0' to retry server '^1'.
15530	Failed to authenticate the user '^0' against the console server '^1'.
15531	The folder '^0' does not exist in Report Center running on '^1'. Queries can only be done if this folder exists.
15532	The query '^0' could not be found in the folder '^1' in Report Center running on '^2'.
15533	Cannot connect to the console server; the host name '^0' is invalid.
15534	Failed to make RPC connection to host '^0' due to '^1'.
15535	Query result '^0' does not have the required '^1' column.
15536	Error reading macros file '^0': ^1
15537	Error reading macros from URL '^0': ^1
15538	The console settings have not been configured properly. Configure the console (a system setting) so that Deployment Manager can access the machine hosting Report Center, and can log in to run the appropriate query.
15539	Failed to access Report Center. Make sure Report Center has been started.
15540	Failed to access the queries from Report Center. The database '^0' is down.
15541	Cannot resolve the console host name '^0'.
15542	Connection refused on host '^0', port '^1'. Please make sure that a tuner is running on that host and management port.
15550	Status from SMTP mailer: ^0 for job '^1'.

Log ID	Log message for Server Management
15555	The task does not contain valid targets. Please verify TMS task Relationships:
15556	The task does not contain valid source (channels). Please verify TMS task Relationships:
15557	The task Id ^0 does not exist in the AR System
15558	The Extended type of channels are not subscribed.
15559	Cannot connect to CMS ARManager Service
15560	Failed to create context on CMS AR Gateway: ^0
15561	Failed to get Task detail. ^0. ^1
15562	Failed to process Task. ^0. ^1
15563	The change Id ^0 does not exist in the AR System
15580	Database connection restored
15581	Database connection dropped
15600	Processed ^0 log-entries from client ^1 in ^2ms
15601	Failed to process log-data from client ^0
15602	Failed to process entry in log-queue
15620	Executed batch ^0 in ^1ms
15630	Disregarding request of type ^0 from client ^1 job ID ^2 as job is not active
15650	Queued ^0 bytes of log-data from client ^1 in ^2ms
15651	Failed to queue logs from client ^0
15700	Internal Error: illegal argument passed into ^0
15701	Error: the specified system command '^0' is invalid. The format should be system,<true false>=<system_command_to_execute_on_endpoint> .
15702	Error: the specified value '^0' is not one that can be converted into a valid command.
15703	Error: the specified value '^0' is an invalid log success ID.
15704	Error: the specified value '^0' is an invalid log failure ID.
15705	Error: the specified value '^0' is an invalid minimum log ID.

Log ID	Log message for Server Management
15706	Error: the specified value '^0' is an invalid maximum log ID.
15707	Error: the specified value '^0' is an invalid timeout value.
15708	Error: the channel URL field and the channel type field are required fields for a channel command. Please supply values for these fields.
15709	Error: invalid channel URL '^0'.
15710	Error: the channel type field is required for a channel command.
15711	Error: if you fill in one log ID, all log IDs must be filled in.
15712	Error: the success log ID field is required if you want to use the 'command response timeout' field.
15713	Error: the success log ID field is required if you want to use the 'wait for exit' field.
15714	Error: the combination of log min '^0' and log max '^1' values are invalid. If either ID is specified, both IDs must be specified. The minimum value must be less than the maximum value.
15715	Error: the specified value '^0' is an invalid value for waitForExit. The value for waitForExit can be either "true" or "false".
15716	Error: the success log ID field is required if you want to use any other log ID fields.
15717	Error: If a Log Id is entered, then please select waitForExit as 'true'.
15718	Error: If you select the Channel state as 'updateFrom' then Channel URL from which to update is required.
15719	Error: CommandThread doRemoteScriptCommand: Error in file download.
16300	Error: the SOAP message contains a badly formed XML document.
16301	Error: the SOAP message does not conform to the SOAP 1.1 specification.
16302	Error: the SOAP message contains an invalid element specification.
16303	Error: a SOAP client fault occurred due to an IO error.
16304	Error: a SOAP client fault occurred. The SOAP message contains one or more XML errors.
16305	Error: a SOAP client fault occurred. The SOAP message does not conform to the SOAP 1.1 specification.

Log ID	Log message for Server Management
16306	Error: a SOAP client fault occurred. The SOAP message contains an invalid element specification.
16307	Error: a SOAP client fault occurred due to an unexpected error.
16308	Error: the SOAP message contains a badly formed XML document.
16309	Error: end tag '^0' has no corresponding start tag.
16310	Error: end tag '^0' does not match start tag '^1'.
16311	Error: invalid element '^0', the first element must be an Envelope.
16312	Error: invalid element '^0', SOAP message contains multiple root elements.
16313	Error: no namespace for header child element '^0'.
16314	Error: no namespace URI for header child element '^0'.
16315	Error: subelement '^0' may not appear inside element '^1', or it has exceeded its maximum number of instances.
16316	Error: element '^0' was closed before meeting its minimum requirements.
16317	Error: element '^0' has an invalid namespace URI '^1'.
16318	Error: element '^0' has no namespace URI.
16500	Internal error: problem getting extended type info for edited object: ^0.
16501	Internal error: dm.obj.type property specified for path '^0' has invalid
16502	Internal Error: form '^0' is missing compulsory argument '^1'
16503	Warning: You tried to access another page. Any changes you have made on this page will not be saved.
16504	You cannot proceed to the desired page specified in this warning because you have canceled out of the edits that produced this warning. Deployment Manager does not store the edit session information of your previous edits.
16505	Internal error: Deployment Manager could not find the necessary navigation information '^0' to redirect you to the desired page.
16506	Before submitting this page, please correct the following errors:
16507	Page to redirect to not defined.

Log ID	Log message for Server Management
16508	Error: Cannot access the default folder specified for your account '^0' due to: ^1. You are being placed in the root folder. Please contact your Deployment Manager administrator to make your default folder available to you.
16509	Can not perform the '^0' operation on some jobs.
16530	Error: the operation cannot be performed because the edit session for that ^0 has been canceled or is otherwise invalid.
16531	Internal Error: Type for object not specified.
16532	Internal error: dm.obj.type property not found for handler '\^0\'.
16533	Internal Error: no parent object session variable for '\^0\'.
16534	Internal Error: session variable for '\^0\'' does not instantiate correct interface class.
16535	Internal Error: parent session variable for '\^0\'' does not instantiate correct interface class.
16536	Internal Error: cannot create edit session copy for '\^0\''
16537	Error: no object to get type info '\^0\''
16538	Error: cannot perform '^0' operation on '\^1\'' due to internal error: '^2'
16539	Error: cannot perform '^0' operation on '\^1\'' due to: '^2'
16540	Internal Error: handler '^0' cannot be initialized because object class '^1' cannot be loaded.
16550	Internal Error: help for integration module '\^0\'' not found.
16551	Internal Error: fail to initialize search database for help for integration module '\^0\''. Error initializing search properties.
16600	Error: attempting to delete a group at an index that does not exist.
16620	Internal Error: missing parameter for job reference operation: '\^0\''
16640	Internal Error: Job '\^0\'' not found.
16641	Can't perform operation '\^0\'' on job due to: '\^1\'' handler errors.
16675	Error in removing group, check if you have requisite permissions.
16676	Please specify a valid host name.
16677	Host and base domain must both be supplied.
16678	Could not connect to LDAP server.

Log ID	Log message for Server Management
16679	No value was specified when adding a user.
16680	The user you wish to add already exists in the user database.
16681	No value was specified when adding an LDAP user or group.
16682	Spaces are not allowed in user names.
16683	No user name was specified.
16684	The user could not be added, perhaps due to inadequate privileges.
16685	Error in creating group: check the Deployment Manager logs.
16686	No name specified.
16687	User does not exist.
16688	Group does not exist.
16689	Error adding the user to the local database: ^0
16690	Passwords do not match.
16691	Setting of password failed.
16692	Spaces are not allowed in group names.
16693	The group you wish to add already exists.
16694	No group name specified.
16695	Removing the admin user from the dmadmins group is not allowed.
16696	User '^0' does not exist.
16697	Could not save group, check privileges.
16698	The interval should be between 5 and 10000 minutes.
16699	Insufficient privileges to edit group.
16700	The Deployment Service URL entered is not formatted as a URL. Please enter the value as <http or https>://<path to SDMClient>
16701	The Macro Variable ^0 is named with the reserved prefix ^1. Please rename the macro variable with a different prefix.
16702	Invalid root directory path.
16703	The user you wish to add already exists.
16704	Specify either command or browse a file, both not allowed.

Log ID	Log message for Server Management
18000	Internal Error: integration module '\^0\' cannot be loaded properly. Check properties.txt configuration for integration module. Possible error in property: '\^1\'.
18001	Error: integration module '\^0\' not found.
18002	Internal Error: integration module lookup not found.
18003	Internal Error: error in reading command-line declaration file (sdm cmdline.txt in main integration module directory) for integration module '\^0\'.
18004	Internal Error: integration module '\^0\' attempt to define a command '\^1\' that is already being defined by the Deployment Manager or another module.
18005	Internal Error: integration module '\^0\' cannot modify a Deployment Manager object command '\^1'. Check sdm cmdline.txt to make sure syntax of an object command is <baseobj>.<entryname>.CMD_<objcommand>.
18006	Error: helper type '\^0\' not found.
18007	Internal error: helper type property '^1' for helper '^0' not found. The integration module developer did not define the property in the module's configuration file.
18008	Error: Failed to initialize integration module '^0' due to: ^1
18009	Error: missing compulsory property '^0' for '^1'
18010	Error: invalid abbreviation for object type for '^0'. Valid values are "tg", "cmd", "dep", "sg", or "sk".
18011	Error: invalid object type for valid parent property: '^0' for '^1'.
18012	Error: missing compulsory property '^0' for integration module '^1'
18013	Error: missing compulsory component '^0' for integration module '^1'
18014	Error: ^0 does not implement ^1: ^2
18015	Internal Error: failed to load integration module main class
18016	Error: unexpected exception while defining command line for integration module ^1: ^2
18017	Error: extended type '^0' from integration module '^1' not installed. Make sure format is in <object type>:<helper type>.
18018	Error: unexpected exception while loading integration module: ^1

Log ID	Log message for Server Management
18019	Error: hidden extended type '^0' from integration module '^1' not recognized. Make sure format is in <object type>:<helper type>.
18050	Integration module '^0' successfully added.
18051	Integration module '^0' successfully installed.
18052	Integration module '^0' successfully removed.
18053	Failed to add integration module '^0'.
18054	Failed to install integration module '^0'.
18055	Failed to remove integration module '^0'.
18056	Internal error: unable to create the integration module lookup.
18057	Internal error: Unexpected error while performing operation on integration module ^0
18058	Internal error: unable to clean up failed installation of ^0. Some malfunctioning helper types remains in the system.
18101	^0 Folder object '^1' created
18102	^0 ServerKeychain object '^1' created
18103	^0 ServerGroup object '^1' created
18104	^0 TaskGroup object '^1' created
18105	^0 Deployment object '^1' created
18106	^0 Folder object '^1' deleted
18107	^0 ServerKeychain object '^1' deleted
18108	^0 ServerGroup object '^1' deleted
18109	^0 TaskGroup object '^1' deleted
18110	^0 Deployment object '^1' deleted
18111	^0 Folder object '^1' renamed to '^2'
18112	^0 ServerKeychain object '^1' renamed to '^2'
18113	^0 ServerGroup object '^1' renamed to '^2'
18114	^0 TaskGroup object '^1' renamed to '^2'
18115	^0 Deployment object '^1' renamed to '^2'
18116	^0 Folder object '^1' moved to '^2'
18117	^0 ServerKeychain object '^1' moved to '^2'

Log ID	Log message for Server Management
18118	^0 ServerGroup object '^1' moved to '^2'
18119	^0 TaskGroup object '^1' moved to '^2'
18120	^0 Deployment object '^1' moved to '^2'
18121	^0 Folder object '^1' copied to '^2'
18122	^0 ServerKeychain object '^1' copied to '^2'
18123	^0 ServerGroup object '^1' copied to '^2'
18124	^0 TaskGroup object '^1' copied to '^2'
18125	^0 Deployment object '^1' copied to '^2'
18126	^0 '^1' Folder object Properties changed: ^2
18127	^0 '^1' ServerKeychain object Properties changed: ^2
18128	^0 '^1' ServerGroup object Properties changed: ^2
18129	^0 '^1' TaskGroup object Properties changed: ^2
18130	^0 '^1' Deployment object Properties changed: ^2
18131	^0 '^1' Folder object Properties Enabled: ^2
18132	^0 '^1' ServerKeychain object Properties Enabled: ^2
18133	^0 '^1' ServerGroup object Properties Enabled: ^2
18134	^0 '^1' TaskGroup object Properties Enabled: ^2
18135	^0 '^1' Deployment object Properties Enabled: ^2
18136	^0 '^1' Folder object Properties Disabled: ^2
18137	^0 '^1' ServerKeychain object Properties Disabled: ^2
18138	^0 '^1' ServerGroup object Properties Disabled: ^2
18139	^0 '^1' TaskGroup object Properties Disabled: ^2
18140	^0 '^1' Deployment object Properties Disabled: ^2
18141	^0 Folder Object Permission Changed: set permission of user '^1' to ^2
18142	^0 KeyChain Object Permission Changed: set permission of user '^1' to ^2
18143	^0 Server Group Object Permission Changed: set permission of user '^1' to ^2

Log ID	Log message for Server Management
18144	^0 Task Group Object Permission Changed: set permission of user '^1' to ^2
18145	^0 Deployment Object Permission Changed: set permission of user '^1' to ^2
18146	^0 Folder Object Permission Changed: set permission of group '^1' to ^2
18147	^0 KeyChain Object Permission Changed: set permission of group '^1' to ^2
18148	^0 Server Group Object Permission Changed: set permission of group '^1' to ^2
18149	^0 Task Group Object Permission Changed: set permission of group '^1' to ^2
18150	^0 Deployment Object Permission Changed: set permission of group '^1' to ^2
18151	^0 Group '^1' Removed from the permission list
18152	^0 Group '^1' Added to the permission list
18153	^0 User '^1' Removed from the permission list
18154	^0 User '^1' Added to the permission list
18155	^0 ServerKeychain object '^1' added to ServerGroup object '^2'
18156	^0 ServerKeychain object '^1' removed from ServerGroup object '^2'
18157	^0 ServerGroup object '^1' added to Deployment object '^2'
18158	^0 ServerGroup object '^1' removed from Deployment object '^2'
18159	^0 TaskGroup object '^1' added to Deployment object '^2'
18160	^0 TaskGroup object '^1' removed from Deployment object '^2'
18161	^0 Server '^1' added to ServerGroup object '^2'
18162	^0 Server '^1' removed from ServerGroup object '^2'
18163	^0 Server Host '^1' added to ServerKeychain object '^2'
18164	^0 Server Host '^1' removed from ServerKeychain object '^2'
18165	^0 Task '^1' added to TaskGroup object '^2'
18166	^0 Task '^1' removed from TaskGroup object '^2'
18167	^0 Task '^1' created to TaskGroup object '^2'

Log ID	Log message for Server Management
18168	^0 Task name '^1' renamed to '^2'
18169	^0 Command '^1' created to Task '^2'
18170	^0 Command '^1' deleted from Task '^2'
18171	^0 Copy of command '^1' created to Task '^2'
18172	^0 Task '^1' command changed to '^2'
18173	^0 Login successful
18174	^0 Login failed
18175	^0 Macros [^1] added to TaskGroup '^2'
18176	^0 Macros [^1] removed from TaskGroup '^2'
18177	^0 Macros [^1] added to Server '^2'
18178	^0 Macros [^1] removed from Server '^2'
18179	^0 Macros [^1] added to ServerGroup '^2'
18180	^0 Macros [^1] removed from ServerGroup '^2'
18181	^0 Task '^1' properties changed:[^2]
18182	^0 Logoff successful
18183	User has been timed out.
18184	^0 '^1' Deployment Object Job Properties changed: ^2
18185	^0 '^1' Deployment Object Job Properties Enabled: ^2
18186	^0 '^1' Deployment Object Job Properties Disabled: ^2
18187	^0 Login successful from TMS
18188	^0 Login failed from TMS

Infrastructure Service

This section lists the log IDs, corresponding log messages, and log severity levels for Infrastructure Service.

Log ID	Log message for Infrastructure Service	Severity level
32009	Tuner update failed	CRITICAL
32044	Tuner update succeeded	AUDIT

Log ID	Log message for Infrastructure Service	Severity level
32060	Tuner Update Manager disabled	AUDIT
32061	Tuner restart required	AUDIT
32062	Tuner restart not allowed	WARNING
32200	Version	AUDIT
32201	Unsupported operating system	WARNING
32203	Tuner must be restarted to install pending update	WARNING
32204	Lock acquired	AUDIT
32205	Lock released	AUDIT
32206	Lock failed	CRITICAL
32207	Lock stale	AUDIT
32210	Verify index started	AUDIT
32211	Verify index done	AUDIT
32212	Verify index failed	CRITICAL
32213	Verify filemap started	AUDIT
32214	Verify filemap done	AUDIT
32215	Verify filemap failed	CRITICAL
32216	Verify ok	AUDIT
32217	Verify file modified	MAJOR
32218	Verify file missing	MAJOR
32219	Verify file needs to be mapped	MINOR
32220	Verify file needs to be merged	MINOR
32221	Verify skipped	AUDIT
32222	Verify cannot read	CRITICAL
32223	Verify cannot write	CRITICAL
32224	Verify file is now a directory	MAJOR
32225	Verify directory is now a file	MAJOR
32226	Verify stale file map entry	MINOR
32227	Verify file is corrupt	AUDIT
32228	Verify exception	CRITICAL

Log ID	Log message for Infrastructure Service	Severity level
32250	Prepare started	AUDIT
32251	Prepare done	AUDIT
32252	Prepare failed	CRITICAL
32253	Prepare added	AUDIT
32254	Prepare not mapped	AUDIT
32255	Prepare excluded	AUDIT
32300	Editor added file	AUDIT
32301	Editor mapped file	AUDIT
32302	Editor unmapped file	AUDIT
32303	Editor unmapped directory	AUDIT
32304	Editor deleted file	AUDIT
32305	Editor cannot unmap a file that is not mapped	WARNING
32306	Editor failed to add file	CRITICAL
32307	Editor failed to map file	CRITICAL
32308	Editor failed to unmap file	CRITICAL
32309	Editor failed to unmap directory	CRITICAL
32310	Editor exception occurred	CRITICAL
32340	Filemap case mismatch	WARNING
32350	Update starting	AUDIT
32351	Update done	AUDIT
32352	Update failed	CRITICAL
32353	No changes necessary	AUDIT
32400	Profile requested	AUDIT
32401	Profile not on the Transmitter	CRITICAL
32402	Profile applied to channel	AUDIT
32403	Profile cannot be applied to channel	MINOR
32404	Profile matches more than one channel	MAJOR
32405	Profile error occurred	MAJOR

Log ID	Log message for Infrastructure Service	Severity level
32406	Profile exists, but marimba.tuner.update.profile property not set	WARNING
32450	Install started	AUDIT
32451	Install done	AUDIT
32452	Install failed	CRITICAL
32453	Install code changed	AUDIT
32454	Created	AUDIT
32455	Updated	AUDIT
32456	Install failed	CRITICAL
32457	Deleted	AUDIT
32458	Delete failed	CRITICAL
32459	Merged	AUDIT
32460	Merge skipped	AUDIT
32461	Merge failed	CRITICAL
32462	Node missing	CRITICAL
32463	Install detected zero-length file	CRITICAL
32464	Install committed	AUDIT
32465	No changes to install	AUDIT
32466	Install unknown type	CRITICAL
32467	Install ignore mode bits	AUDIT
32468	Segment missing on the Transmitter	CRITICAL
32469	Segment corrupt on the Transmitter	CRITICAL
32470	Check failed	CRITICAL
32500	Schedule initialized	AUDIT
32501	Schedule changed	AUDIT
32502	Schedule invalid	MAJOR
32503	Schedule delayed	AUDIT
32520	Restarting tuner	AUDIT
32521	Restarting channel	AUDIT

Log ID	Log message for Infrastructure Service	Severity level
32550	Mapping started	AUDIT
32551	Mapping done	AUDIT
32552	Mapping failed	CRITICAL
32600	Failed to delete	CRITICAL
32601	Failed to save	CRITICAL
32602	File missing	CRITICAL
32603	Failed to open	CRITICAL
32604	Failed to clear	CRITICAL
32605	Failed to create directory	CRITICAL
32606	Disk space (required/available)	AUDIT
32607	Not enough disk space (required/available)	CRITICAL
32608	Failed to check disk space	WARNING
32609	Cannot check disk space on older tuner	WARNING
32667	Transmitter exception	CRITICAL
32668	Requested profile has not been published	WARNING
32700	No corruption found in Tuner workspace. Nothing to repair.	AUDIT
32701	Corruption found in Tuner workspace. Will repair when tuner restarts.	AUDIT
32702	Restarting tuner for workspace repair.	AUDIT
48100	ADP Getfiles	
48101	HTTP post	
48102	HTTP Get	
48103	HTTP Connec	
48150	Protocol Error	
48151	SSL Setup Error	
48152	Denial Error	
48153	Process Error	
48154	IOException while processing request	MINOR

Log ID	Log message for Infrastructure Service	Severity level
48221	Changing maximum cache size to	AUDIT
48222	Changing cache low water mark to	AUDIT
48230	Started cache collection at current cache size of	AUDIT
48231	Stopped cache collection at current cache size of	AUDIT
48232	No room in cache to insert file of length	AUDIT
48240	Refreshing cache with channel group	AUDIT
48241	Refreshing cache with car file	AUDIT
48250	Property added	AUDIT
48251	Property deleted	AUDIT
48252	Property modified	AUDIT
48405	Java 2 VM required, current VM version	CRITICAL
48410	Error while refreshing port properties	MAJOR
48411	Error while refreshing port properties	WARNING
48412	Error with water mark specification	MINOR
48414	Could not resolve outgoing host address	MAJOR
48430	Exception while processing request from	MINOR
48431	Exception occurred while retrieving files from transmitter for URL	MINOR
48440	Could not open channel URL	MINOR
48441	Got invalid transmitter reply while updating channel	MINOR
48442	Exception while refreshing cache with URL	MINOR
48443	Exception while refreshing from car file	MINOR
48450	Unexpected HTTP reply from transmitter	MAJOR
48451	No XML data received from transmitter	MAJOR
48452	Exception occurred with making an XML listing to the transmitter	MAJOR
48500	Sending status report	AUDIT
48701	Got access to tuner storage: Tuner storage is accessible through gfs	INFO

Log ID	Log message for Infrastructure Service	Severity level
49100	failed to get the config object	
49103	UDP Broadcast Server started successfully	
49104	MESH service added as an observer to the UDP Broadcast server	
49105	UDP Broadcast initialization error	
49107	Tuner storage error	
49108	Tx file download failed for some reason other than a missing DIFF	
49109	MESH is enabled for this tuner	
49110	MESH is disabled for this tuner	
49111	Error in MESH initialization	
49112	ADP File request	
49113	Peer list is corrupted or not initialized	
49114	No more peers in the list left	
49115	ADP Request finished	
49116	New files installed into channel	
49117	MESH was enabled, but getfiles info missing	
51000	Health & Monitoring Report: Total time for request	INFO
51001	Health & Monitoring Report: Total time on connection	INFO
51002	Health & Monitoring Report: Size of stats sent	INFO
51003	Tuner out of network, will miss sending stats report	WARNING
51004	Tuner H&M Module started	INFO
51005	Sending Tuner's H&M stats report	INFO
51006	Missed report sent	INFO
51007	Next report scheduled to be sent at	INFO
51008	Stats report was not sent successfully	CRITICAL

Log ID	Log message for Infrastructure Service	Severity level
51009	Problem while processing plugin reply	MAJOR
51010	Unable to read previous saved stats report	WARNING

Infrastructure Status Monitor

This section lists the log IDs, corresponding log messages, and log severity levels for Infrastructure Status Monitor.

Log ID	Log message for Infrastructure Status Monitor	Severity level
52101	Application starting	INFO
52102	Application stopping	INFO
52151	Added property	AUDIT
52152	Updated property	AUDIT
52153	Updated group	AUDIT
52154	Restored formula group values	AUDIT
52155	Published formula to plugin	AUDIT
52156	Failed to publish formula	MAJOR
52201	Created notification	AUDIT
52202	Failed to create notification	MINOR
52203	Updated notification	AUDIT
52204	Failed to update notification	MINOR
52205	Deleted notification	AUDIT
52206	Failed to delete notification	MINOR
52207	Sent notification	INFO
52208	Failed to send notification	MAJOR
52209	Notification scheduled at	INFO
52210	Failed to Schedule notification	MINOR
52211	Next invocation of notification	INFO
52212	Notification not sent	INFO

Inventory

This section includes logging information for the following components of Inventory:

- “Scanner Service” on page 371
- “Inventory plug-in” on page 374

For logging information for Report Center, see “Report Center” on page 401.

Scanner Service

This section lists the log IDs, corresponding log messages, and log severity levels for Scanner Service.

Log ID	Log message for Scanner Service	Severity level
6000	Starting inventory scan	INFO
6001	Failed to subscribe to scanner extension channel	MAJOR
6002	Failed writing the inventory scan to the file system	MINOR
6003	Failed writing the scanner extension channel’s inventory scan to the file system	MAJOR
6004	Internal error while scanning	MAJOR
6005	Native and tuner scans failed with codes	CRITICAL
6006	Native scan failed with code	MAJOR
6007	Tuner scan failed with code	MAJOR
6008	All scans completed successfully	AUDIT
6009	Copy of executable failed	MINOR
6010	Failed parsing native scan	MAJOR
6011	Failed loading the cached scan report	MAJOR
6012	Failed loading the inventory tree from current scan	MAJOR
6013	Failed diffing the old and new scans	MAJOR
6014	Failed running the scanner extension channel	MAJOR
6015	Done executing scanner extension channels	AUDIT
6016	Scan progress bar cancelled	MINOR

Log ID	Log message for Scanner Service	Severity level
6017	Failed to read from scanner output file	MAJOR
6018	The native process has timed out	MAJOR
6019	The native process has been terminated	MAJOR
6020	Failed to run the native scan	MAJOR
6021	Failed to scan the transmitter	MINOR
6022	Found corrupt license	MINOR
6023	Scanner extension channel URL missing	MAJOR
6024	Failed to uncompress cached report	MAJOR
6025	Failed to read cached report	MAJOR
6026	Sending full report to the plug-in	AUDIT
6027	File Locked. There must be a sharing violation	MAJOR
6028	Scanner extension channel has completed	INFO
6029	Stopping scanner extension channel because it exceeded timeout	MAJOR
6030	Error while subscribing scanner extension channel	MAJOR
6031	Subscribing scanner extension channel exceeded timeout	MAJOR
6032	Warning: scanner extension channel already running; starting another instance	WARNING
6033	Error while updating or running scanner extension channel	MAJOR
6034	Could not start the scanner extension channel because it is not trusted	MAJOR
6035	Malformed timeout value obtained from marimba.inventory.customscanner.timeout	WARNING
6036	Restarting channel	AUDIT
6037	Software usage not supported on this system	AUDIT
6038	Software usage monitor log files harvested	AUDIT
6039	Scan component completed	AUDIT
6040	Updated next scheduled wakeup	AUDIT
6041	Channel start type	AUDIT

Log ID	Log message for Scanner Service	Severity level
6042	Failed to load native library	MINOR
6043	Checked next scheduled wakeup	AUDIT
6044	Failed to update scanner extension channel	MINOR
6045	Updating scanner extension channel exceeded timeout	MAJOR
6046	Software usage monitor started at tuner startup	AUDIT
6047	Software usage monitor not started at tuner startup because there is no license for it	AUDIT
6048	Software usage monitor not started at tuner startup because it is not enabled	AUDIT
6100	Sending the scan data to the plug-in	INFO
6101	Sending cached copy	INFO
6102	Failed getting cached file	MINOR
6103	Failed getting cached diff file	MINOR
6104	Plug-in requested full report	AUDIT
6105	HTTP error while talking to plug-in	MAJOR
6106	Exception occurred while sending report	MAJOR
6107	Scan report sent successfully	AUDIT
6108	Channel URL on master	AUDIT
6109	Repeater URL	AUDIT
6110	XML file not found	MAJOR
6111	Failed to parse index file	MAJOR
6112	Server Management scan succeeded	AUDIT
6113	PDA scan succeeded	AUDIT
6114	Server Management scan failed	MAJOR
6115	PDA scan failed	MAJOR
6116	Failed to parse data file	MAJOR
6117	channel version	AUDIT
6118	Resetting cache	AUDIT
6119	Added report to the cache	AUDIT

Log ID	Log message for Scanner Service	Severity level
6120	Error: invalid value for cache capacity	MINOR
6121	Report cache is disabled	AUDIT
6122	Report cache is at capacity; scan will not be added to the cache	AUDIT
6123	Inventory Plug-in is down for maintenance; scanner will retry	MINOR
6125	Scanner is sending report to the Transmitter shown in the log	AUDIT
6139	Security Benchmarks scan not supported in non-windows OS	MINOR

Inventory plug-in

This section lists the log IDs, corresponding log messages, and log severity levels for the Inventory plug-in for the Scanner Service channel.

Log ID	Log message for Inventory plug-in	Severity level
6500	Configurator starting in master mode	AUDIT
6501	Configurator starting in repeater mode. Will do inserts	AUDIT
6502	Configurator starting in slave mode. Will not do inserts	AUDIT
6503	Configurator initialized successfully	AUDIT
6504	Configurator stopping	AUDIT
6505	Using configurator with index	AUDIT
6506	Configurator not initialized	MAJOR
6510	Going to get checksums	AUDIT
6511	Number of checksums retrieved so far	AUDIT
6512	Total number of checksums retrieved	AUDIT
6513	Requesting checksums from master	AUDIT
6514	Retrieved checksums from master	AUDIT
6515	Sent checksums to	AUDIT

Log ID	Log message for Inventory plug-in	Severity level
6520	Forwarded report to master/mirror	AUDIT
6521	Requested full scan for this machine	AUDIT
6522	Connected to database	AUDIT
6523	Inserted scan report in	INFO
6524	Property added	AUDIT
6525	Property deleted	AUDIT
6526	Property modified	AUDIT
6527	Report queued	INFO
6528	Differential report queued	INFO
6529	Report queue initialized. Number of existing reports	AUDIT
6530	Using driver	AUDIT
6531	Plug-in is not enabled	AUDIT
6532	Native scan timed out for machine. Only status and scan time updated	AUDIT
6533	Plug-in report insertion is disabled	AUDIT
6534	Report not inserted	AUDIT
6535	Not going to forward reports to any master/mirror	AUDIT
6536	Plug-in reports will be forwarded to master/mirror	AUDIT
6537	Requested diff scan for this machine	AUDIT
6600	Problem reading thread settings. Will use default min = 2 and max = 5	MINOR
6610	Could not setup connection with database server	CRITICAL
6611	Problem connecting to database server. Trying again	CRITICAL
6612	Could not find class for JDBC driver	CRITICAL
6613	Dropped scan report	MINOR
6614	Timed out waiting for connection. Trying again	CRITICAL
6620	Error storing checksums from master	MAJOR
6621	Error while sending checksums to repeater	MAJOR

Log ID	Log message for Inventory plug-in	Severity level
6622	HTTP error while fetching checksums from master	MAJOR
6623	Error connecting to database while updating checksums	MAJOR
6630	Error storing checksum in hash	MINOR
6640	Failed to apply diff because of error in checksums	MINOR
6641	HTTP Error connecting to master	MAJOR
6642	Error forwarding log request to master	MAJOR
6650	Scan report was not successfully inserted	MAJOR
6651	Dropping scan report because scan queue is full	MAJOR
6652	Dropping scan report since it is already being processed by another thread	MAJOR
6653	Could not rename report	MAJOR
6660	Invalid version for HTTP request	MAJOR
6661	Unknown request type	MINOR
6662	Exception occurred while reading from slave	MINOR
6663	Hash is still uninitialized	MINOR
6664	Disk hash copier has not started yet	MINOR
6665	Failed copying file for hash	MINOR
6666	Database exception	MAJOR
6667	Error updating the plug-in cache	MINOR
6668	File system access denied	CRITICAL
6669	Failed to add a scan to the scan queue	MINOR
6670	Failed to load a scan from the scan queue	MINOR
6671	Live database connection timed out	MAJOR
6672	Over-writing older report for machine	WARNING
6673	Giving up on report	MAJOR
6674	Scheduled to retry scan report	AUDIT
6675	Could not forward report due to unexpected checksum mismatch	WARNING
6676	Failed to commit	MAJOR

Log ID	Log message for Inventory plug-in	Severity level
6677	Failed to rollback	MAJOR
6678	Incompatible database schema version	CRITICAL
6679	Failed to uncompress scan report	MAJOR
6680	Deleting checksum from cache for mac	WARNING
6681	Inventory plugin not configured yet	CRITICAL
6682	Unable to establish database connection	CRITICAL
6683	Unable to determine schema version	CRITICAL

Logging

This section includes logging information for the following components of centralized logging:

- “Logging Service” on page 377
- “Logging plug-in” on page 378

Logging Service

This section lists the log IDs, corresponding log messages, and log severity levels for Logging Service.

Log ID	Log message for Logging Service	Severity level
24000	Logging service started logging	AUDIT
24001	Logging service stopped logging	AUDIT
24002	Failed to send logs to plugin	MAJOR
24003	Failed to create temporary file	CRITICAL
24004	Failed to compress log file	CRITICAL
24005	Failed to connect to the transmitter	CRITICAL
24006	Failed to open a plugin connection	MAJOR
24007	Failed to send logs. Server disk is full	MAJOR

Logging plug-in

This section lists the log IDs, corresponding log messages, and log severity levels for the Logging plug-in.

Log ID	Log message for Logging plug-in	Severity level
24500	Logger plugin starting in master mode	AUDIT
24501	Logger plugin starting in repeater mode. Will do inserts	AUDIT
24502	Logger plugin starting in slave mode. Will not do inserts	AUDIT
24503	Logger plugin initialized successfully	AUDIT
24504	Logger plugin stopping	AUDIT
24505	Forwarded report to master	AUDIT
24506	Property added	AUDIT
24507	Property deleted	AUDIT
24508	Property modified	AUDIT
24509	Connected to database	AUDIT
24510	Log insert successful	AUDIT
24511	Closed database connection	AUDIT
24512	Using driver	AUDIT
24513	Done processing log report	AUDIT
24514	Log report already exists	AUDIT
24515	Log report queued	AUDIT
24550	Masters queue is full, suspending forwarding	AUDIT
24551	Forwarding resumed	AUDIT
24600	Could not setup connection with database server	CRITICAL
24601	Error while forwarding logs	MAJOR
24602	Unknown request type	MINOR
24603	Network Error	MINOR
24604	Problem connecting to database server - trying again	WARNING

Log ID	Log message for Logging plug-in	Severity level
24605	Could not find class for JDBC driver	CRITICAL
24606	Plugin is not initialized	WARNING
24607	Scheduled to retry log report	MINOR
24608	Giving up on log report	MAJOR
24609	Error while creating log queue	CRITICAL
24610	Http error while forwarding logs	MAJOR
24611	Failed to load report from the log queue	WARNING
24612	Log report was not processed	MAJOR
24613	Log queue disk is full	MAJOR
24614	Can't get a database-connection	MAJOR
24615	Failed to rollback	MINOR
24701	Bad protocol or version	
24702	Could not initialize SNMP manager	
24703	SNMP manager initialized properly	
24704	Database connection error	
24705	Plug-in node mapping error	
24706	Database error while performing some operation	
24707	Error while initializing SNMP on the given port, now retrying after a delay of 10 seconds	

Marimba Migration Module

This section lists the log IDs, corresponding log messages, and log severity levels for the Marimba Migration Module.

Log ID	Log message for Marimba Migration Module	Severity level
31000	Migration Module started	AUDIT
31001	Migration Module stopped	AUDIT
31002	Failed to subscribe channel from the transmitter	MAJOR
31003	Channel subscribed successfully	AUDIT

Log ID	Log message for Marimba Migration Module	Severity level
31004	Channel was deleted from tuner workspace	MINOR
31005	Migration tool type	AUDIT
31006	Failed to execute command	MAJOR
31007	Deployment Checker started	AUDIT
31008	Deployment Checker disabled	AUDIT
31009	Deployment Checker ticked	AUDIT
31010	Cannot get user information	MINOR
31011	User denied access. Primary admin access required	MINOR
31020	Unexpected HTTP reply from transmitter	MAJOR
31021	No XML data received from transmitter	MAJOR
31022	Exception occurred with making an XML listing to the transmitter	MAJOR
31030	Configuration settings saved	AUDIT
31040	Deployment started	AUDIT
31041	Deployment deleted	AUDIT
31042	Deployment created	AUDIT
31043	Deployment schedule missed	WARNING
31050	Task created	AUDIT
31051	Task edited	AUDIT
31052	Task deleted	AUDIT
31060	Group created	AUDIT
31061	Group edited	AUDIT
31062	Group deleted	AUDIT
31070	Active Directory not set up	AUDIT
31071	LDAP connection failed	AUDIT
31072	Failed to delete machine account because of an LDAP exception	AUDIT
31073	Machine account was not deleted because it could not be found in Active Directory	AUDIT

Log ID	Log message for Marimba Migration Module	Severity level
31074	Machine has an unknown domain and was not deleted from Active Directory	AUDIT
31075	Deleted machine account from Active Directory	AUDIT
31076	LDAP has been connected	AUDIT
31077	LDAP has been disconnected	AUDIT
31078	LDAP settings has changed	AUDIT

Patch Management

This section includes logging information for the following components of Patch Management:

- “Patch Manager” on page 381
- “Patch Service” on page 386

Patch Manager

This section lists the log IDs, corresponding log messages, and log severity levels for Patch Manager.

Log ID	Log message for Patch Manager	Severity level
41500	Patch Manager main initialized	AUDIT
41501	Patch Manager main disposed	AUDIT
41502	Failed to initialize remote-admin manager	MAJOR
41503	Reporting connection pool is not available from CMS: make sure it is configured.	CRITICAL
41504	Report Center is not installed or is not running on this console server. Suspending Patch Manager. Verify that the Report Center channel is installed and running on this console server; the Patch Manager channel starts automatically.	CRITICAL
41505	Report Center is now detected to be installed and running on this console server. Patch Manager will now resume.	AUDIT

Log ID	Log message for Patch Manager	Severity level
41506	The patch database schema is either not installed or incompatible. Use Schema Manager to install the patch database schema.	CRITICAL
41507	CMS default reporting database has changed. Patch Manager will now re-initialize.	WARNING or AUDIT
41509	Invalid version of Report Center is installed. Report Center 6.5 or later is required.	CRITICAL or AUDIT
41510	Failed to get configuration properties from the plug-in	MAJOR or AUDIT
41511	Invalid plug-in URL	MAJOR or AUDIT
41512	Query is filtered: get machines with the patch	AUDIT
41513	Query is not filtered: get machines with the patch	AUDIT
41514	Query is filtered: get machines without the patch	AUDIT
41515	Query is not filtered: get machines without the patch	AUDIT
41516	Report Center is installed but not configured properly on this console server. Log in to Report Center and make sure it is properly configured; you should then be able to run queries.	CRITICAL or AUDIT
41520	Manual repository update completed	AUDIT
41521	Repository updated with errors	MINOR
41522	Repository update successful	AUDIT
41530	Check Task—patches added to dynamic patch group to send mail	AUDIT
41531	Check Task—No patches added to dynamic patch group to send mail	WARNING or AUDIT
41532	Non-existent patch group	WARNING
41533	Notification Task—update in repository for notification	AUDIT
41534	Notification Task—no update in repository for notification	WARNING or AUDIT
41535	Query Task - patches added to dynamic patch group	AUDIT

Log ID	Log message for Patch Manager	Severity level
41536	Query Task - no patches added to dynamic patch group	WARNING
41540	Failed to get an LDAP connection	MINOR or AUDIT
41571	Starting publish operation on patch group	AUDIT
41572	Failed to download patch	MINOR or AUDIT
41573	Master transmitter is down	MAJOR or AUDIT
41574	Invalid publish credentials were entered	MAJOR
41575	Successfully published patch group	AUDIT
41576	Failed to publish patch group	MAJOR
41577	Publish group operation was canceled for patch group	AUDIT
41650	Patch Service plug-in URL selected	AUDIT
41651	Saved copy of original Patch Service plug-in properties to	AUDIT
41652	Unknown Patch Service plug-in URL	AUDIT
41653	Received Patch Service plug-in configuration for	AUDIT
41654	Configuration origin is database and returns read-only configuration properties	AUDIT
41655	Patch Service plug-in properties displayed from	AUDIT
41656	Canceled Patch Service plug-in configuration changes made to	AUDIT
41657	Preview Patch Service plug-in configuration changes	AUDIT
41658	Preview of Patch Service configuration changes displayed	AUDIT
41659	Successfully published Patch Service configuration to	AUDIT
41660	Saved Patch Service configuration changes to database for	CRITICAL or AUDIT

Log ID	Log message for Patch Manager	Severity level
41661	Failed to publish Patch Service configuration to	CRITICAL or AUDIT
41662	Cancel Patch Service configuration preview	AUDIT
41663	Repository configuration properties displayed from URL	AUDIT
41664	Windows Patch Source configuration properties displayed from URL	AUDIT
41666	Red Hat Enterprise Linux Patch Source configuration properties displayed from URL	AUDIT
41667	Cancel repository configuration changes	AUDIT
41668	Successfully published repository configuration properties for Windows platform to	AUDIT
41669	Failed to publish repository configuration properties for Windows platform to	CRITICAL or AUDIT
41672	Successfully saved repository configuration properties for Windows platform to database	AUDIT
41674	Successfully saved repository configuration properties to database	AUDIT
41675	Cancel repository configuration preview	AUDIT
41676	SQL exception in getting patch details for Patch ID	CRITICAL
41677	Populated the customize details form with patch data for Patch ID	AUDIT
41678	Add a new references row	AUDIT
41679	Successfully saved the customized patch details for Patch ID	AUDIT
41680	SQL error while saving customized patch details for Patch ID	CRITICAL
41681	Cancel customize patch details	AUDIT
41682	Populate drop-down fields based on selected platform	AUDIT
41683	Successfully saved the custom patch details	AUDIT
41684	Failed to save the custom patch details	CRITICAL or AUDIT

Log ID	Log message for Patch Manager	Severity level
41685	Added affected operating system	AUDIT
41686	Added affected application	AUDIT
41687	Added references row	AUDIT
41688	Removed references row	AUDIT
41689	Added affected systems file row	AUDIT
41690	Removed affected systems file row	AUDIT
41691	Added affected registry entry row	AUDIT
41692	Removed affected registry entry row	AUDIT
41693	Before setting new values to preview configuration	AUDIT
41694	Before setting values from preview configuration to form for action from Preview	AUDIT
41695	Loading default values	AUDIT
41696	Loading configured values from repository whose master transmitter is	AUDIT
41697	Successfully published repository configuration properties for Linux platform to	AUDIT
41698	Failed to publish repository configuration properties for Linux platform to	CRITICAL or AUDIT
41699	Successfully saved repository configuration properties for Linux platform to database	AUDIT
41800	Patch repository update initiated	AUDIT
41810	Publish operation initiated for patch group	AUDIT
41811	Revert to last published state for patch group	AUDIT
41812	Delete patch group	AUDIT
41813	Removing patches	AUDIT
41814	Setting patch action	AUDIT
41830	Static patch group created	AUDIT
41831	Dynamic patch group created	AUDIT
41832	Dynamic patch group created	AUDIT
41833	Static patch group created	AUDIT

Log ID	Log message for Patch Manager	Severity level
41834	Duplicate static patch group created	AUDIT
41835	New patch group information saved for patch group	AUDIT
41836	Added selected patches to selected patch groups	AUDIT
41837	Added pre-installation script	AUDIT
41838	Added post-installation script	AUDIT
41839	Adding scripts to	AUDIT
41840	Clearing pre-installation script	AUDIT
41841	Clearing post-installation script	AUDIT

Patch Service

This section lists the log IDs and corresponding log messages for Patch Service.

Log ID	Log message for Patch Service	Severity level
41100	Patch Start	INFO
41101	Patch Done	INFO
41102	Patch Error	MAJOR
41103	Patch Status	AUDIT or WARNING
41104	Service Error	WARNING or CRITICAL
41105	Service Succeeded	INFO
41106	Service Succeeded - Resume Needed	INFO
41107	Service Resuming	INFO
41108	Patch Service will reboot the machine	INFO
41109	Service failed - Resume Needed	MAJOR
41110	Security Status	AUDIT, INFO, or WARNING

Log ID	Log message for Patch Service	Severity level
41111	Security Error	WARNING or MAJOR or CRITICAL
41112	Initialization Error	AUDIT, INFO, or MAJOR
41120	Scan Status	AUDIT, INFO, or WARNING
41121	Scan Error	WARNING or MAJOR or CRITICAL
41125	Algorithm Status	INFO
41130	Download Status	AUDIT
41131	Download Error	WARNING or CRITICAL
41138	Preinstall Status	INFO
41139	Preinstall Error	MAJOR
41140	Install Status	AUDIT, INFO, WARNING, or CRITICAL
41141	Install Error	WARNING or MAJOR or CRITICAL
41142	Time in minutes before reboot	INFO
41143	Maximum time for snooze	INFO

Policy Management

This section includes logging information for the following components of Policy Management:

- “Policy Service” on page 388
- “Policy Manager” on page 392
- “Policy Service plug-in” on page 394

Policy Service

This section lists the log IDs, corresponding log messages, and log severity levels for Policy Service.

Log ID	Log message for Policy Service	Severity level
8500	Subscription service started	INFO
8501	Subscription service stopped	INFO
8510	Setting channel state to silent	INFO
8511	Marking channel as subscriber	INFO
8512	Setting channel state	INFO
8513	Removing subscription for channel	INFO
8514	Changing update URL for channel	INFO
8515	Marking channel as persistent	INFO
8520	Adding an entry in the start menu for channel	INFO
8521	Adding a desktop shortcut for channel	INFO
8522	Number of retries left	INFO
8523	Resolving channel conflict, accepting	INFO
8524	Setting channel/tuner property	INFO
8525	Channel was not subscribed at the end of the maximum waiting time	INFO
8526	System is being rebooted	INFO
8527	Service is scheduled to process next event at	INFO
8528	Service will update the channel now	INFO
8529	Service will verify/repair the channel now	INFO
8530	Service no longer manage channel's update, restore previous schedule	INFO
8531	Service no longer manage channel's verify/repair, restore previous schedule	INFO
8532	Service done updating channel	INFO
8532	Service done updating channel	INFO
8533	Service done verify/repair channel	INFO

Log ID	Log message for Policy Service	Severity level
8534	Process this channel as ASAP channel according to schedule	INFO
8535	The channel's next running time is	INFO
8536	Service is started in an "Update Now" mode	INFO
8537	Service will apply this policy only:	INFO
8538	Service is restarted after update in "Update Now" mode	INFO
8539	Service is restarted after update and will only apply for:	INFO
8540	Received new policy while an instance is already running.	INFO
8541	Installing new policy received during previous run.	INFO
8542	New policy received during previous run was not installed after the timeout.	MAJOR
8543	Reapplying new policy received during previous run.	INFO
8544	Skipping verification of policy application as there's a new policy to be installed. Current retry count:	INFO
8545	Provisioning machine. Setting provision policy groups:	INFO
8546	Service successfully provisioned machine.	INFO
8548	System forcing packaged application immediate reboot	INFO
8549	Setting post provisioning auto start channel:	INFO
8550	Setting machine domain	INFO
8551	Setting user domain	INFO
8552	Setting machine name	INFO
8553	Setting user DN	INFO
8554	Setting machine DN	INFO
8555	Inventory Service started successfully	INFO
8556	Channel started successfully	INFO

Log ID	Log message for Policy Service	Severity level
8557	The tuner property marimba.tuner.nt.reflect.username=false. Policy Service will use the tuner user name.	INFO
8598	Service successfully installed the policies	INFO
8599	Service failed to install the policies	MAJOR
8600	Attempted to set channel to an unknown state	MINOR
8601	Error subscribing to channel	MINOR
8603	Error updating the channel	MINOR
8604	Security Exception while loading mrbasubscription.dll	MAJOR
8605	UnsatisfiedLinkError while loading mrbasubscription.dll	MAJOR
8606	IOException while loading mrbasubscription.dll	MAJOR
8608	Could not read segments.xml file for resolving Segment Selection	MINOR
8609	No such channel to update	MINOR
8610	No such channel to verify/repair	MINOR
8611	Subscription Plugin older than 5.0 is not supported by this version of service. Please upgrade the Subscription Plugin.	MINOR
8612	Could not read channel file mrbsubscription.dll	MINOR
8613	WARNING:DLL mrbsubscription.dll may be locked and new version will not take effect till tuner is restarted	MINOR
8614	cannot update because channel does not exist	INFO
8615	cannot update because channel does not have an update schedule	INFO
8616	cannot update because channel is in AVAILABLE state	INFO
8617	cannot verify because channel does not exist	INFO
8618	cannot verify because channel does not have a verify schedule	INFO

Log ID	Log message for Policy Service	Severity level
8619	cannot verify because channel is not an app packaged channel	INFO
8620	cannot parse marimba.subscription.varytime	INFO
8621	While resolving channel title, cannot parse XML from Transmitter	MINOR
8622	While resolving channel title, cannot process the channel or segment element for channel	MINOR
8623	Cannot parse config.xml. Turn on the DEBUG flag to obtain config_debug.xml from the data directory	MAJOR
8624	Cannot process an element from the config.xml	MAJOR
8625	Config.xml not received from the plug-in. Please check the plug-in's log on the Transmitter for errors.	MAJOR
8626	Invtree file that is passed to -computeCompliance doesn't exist.	MAJOR
8627	Subscription failed to update with the exception:	MAJOR
8628	Unable to open config.xml file. File may not be present or the plug-in may not be published Policy Service.	MAJOR
8629	Error while closing the IO stream.	MAJOR
8630	Skipped writing group membership details and computing compliance since unable to open the config.xml.	MAJOR
8631	Finish waiting for channel	INFO
8632	Start observing NCP channel	INFO
8633	Start observing channel	INFO
8634	Start update of channel	INFO
8635	Wait for channel to Subscribe	INFO
8636	Failed to apply provision group, provisioning license not enabled	INFO
8637	Subscription update schedule expired, trying to restore channel's schedule	INFO

Log ID	Log message for Policy Service	Severity level
8638	Subscription verify/repair schedule expired, trying to restore channel's schedule	INFO
8700	Not installed for local user	INFO

Policy Manager

The following table lists the log IDs, corresponding log messages, and log severity levels for Policy Manager.

Note: A number enclosed by curly braces ({x}) indicates that the name of a specific object, such as the name of a target or package, will be inserted in that location of the log message. For example, in the log message “Entire subscription (including properties and packages) deleted for target: {0}.” {0} will be replaced by a target name.

Log ID	Log message for Policy Manager	Severity level
8700	User logged in.	AUDIT
8701	User failed to log in.	AUDIT
8704	Entire policy (including properties and packages) deleted for target: {0}.	AUDIT
8706	User logged out.	AUDIT
8708	Package(Primary State,Secondary State): {0} Modified for target: {1} Schedules (primary,secondary,update,repair): {2}	AUDIT
8709	Package(Primary State,Secondary State): {0} Deleted for targets: {1}	AUDIT
8711	Configured the Policy Service plug-in settings for plug-in: {0}	AUDIT
8713	Set blackout period for target: {0}	AUDIT
8714	Set \${tuner} and package properties for target: {0}	AUDIT
8715	Set \${transmitter} permissions for target: {0}	AUDIT
8716	Set Policy Service schedule for target: {0}	AUDIT

Log ID	Log message for Policy Manager	Severity level
8719	User failed to log out.	AUDIT
8722	Conflicts while saving properties for target: {0}. The conflicts encountered were the following: {1}	AUDIT
8724	Push Deployment id {0} Started	AUDIT
8725	Push Deployment id {0} Completed	AUDIT
8727	Push Deployment id {0} Stopped	AUDIT
8728	Push Deployment id {0} Retry	AUDIT
8729	Deployment Manager settings changed, Policy Manager reinitialized the Deployment manager object.	MAJOR
8730	Patch Service URL not found. The immediate policy update will only issue the Policy Service update command.	AUDIT
8734	Connection to the Global Catalog failed.	MAJOR
8738	Invalid search filter.	WARNING
8739	ACL storage failed.	MAJOR
8740	Add members failed.	MAJOR
8741	Add members succeeded.	MAJOR
8742	Internal exception occurred.	MAJOR
8744	Collection refreshed successfully.	MAJOR
8745	Refresh failed for collection.	MAJOR
8746	Collection preview succeeded.	MAJOR
8747	Preview failed for collection.	MAJOR
8748	Collection deleted successfully.	MAJOR
8756	LDAP connection failed.	MAJOR
8757	Page size should not be a string.	MAJOR
8758	Remove task failed.	MAJOR
8759	Report centre not configured.	MAJOR
8760	LDAP query collection creation failed.	MAJOR
8763	LDAP query collection schedule is not set.	CRITICAL

Log ID	Log message for Policy Manager	Severity level
8764	LDAP query collection user not found in LDAP.	MAJOR
8765	LDAP query collection null query string.	WARNING

Policy Service plug-in

This section lists the log IDs, corresponding log messages, and log severity levels for the Policy Service plug-in.

Log ID	Log message for Policy plug-in	Severity level
8100	Subscription plugin started	AUDIT
8101	Subscription plugin stopped	AUDIT
8102	Making connection to LDAP server	AUDIT
8103	Initialized plugin LDAP configuration	AUDIT
8104	Processed subscription request successfully. Following seven numbers indicate the time required in (ms) for searching (machine,user,machine domain, user domain, groups,subscriptions,total) for machine/user	AUDIT
8105	Processing request, machine name	AUDIT
8109	Name of the Transmitter entry used from the LDAP Server mapping file is	AUDIT
8110	The plugin is not processing requests, the Subscription Config has indicated that the plugin is in the offline mode	AUDIT
8111	The plugin could not read the contents of the subscription container, check plugin user permissions	AUDIT
8112	OS Provisioning DN's received from endpoint	AUDIT
8200	Subscription Plugin has not been initialized.	CRITICAL
8210	Error connecting to LDAP server	MAJOR
8211	Error executing query on LDAP server	MAJOR
8212	Naming error while executing query	MAJOR
8213	IO error while executing query	MAJOR

Log ID	Log message for Policy plug-in	Severity level
8214	Runtime error while executing query	MAJOR
8216	Invalid Transmitter name format (host:[port]) in LDAP Server mapping file on line number	MAJOR
8217	Invalid LDAP server format (host:[port]) in LDAP Server mapping file on line number	MAJOR
8218	Invalid SSL option (allowed valued are {true or false}) on line number	MAJOR
8219	Invalid key in LDAP Server mapping file on line number	MAJOR
8220	Duplicate key in LDAP Server mapping file on line number	MAJOR
8221	Error while reading LDAP Server mapping file	MAJOR
8222	Base DN was not specified in the LDAP Server mapping for transmitter	MAJOR
8223	Bind DN was not specified in the LDAP Server mapping along with the password on line	MAJOR
8224	Password was not specified in the LDAP Server mapping along with the Bind DN on line	MAJOR
8225	LDAP server or domain (if using AD without auto-discovery) was not specified in the LDAP Server mapping for transmitter	MAJOR
8226	There is a problem with loading the credentials supplied during the publish process. This is an internal error. Please report this problem to the software distributor.	MAJOR
8227	The current domain of the plugin cannot be obtained. This is needed to resolve the full name of the client endpoints. Please determine that the machine hosting the plugin belongs to an Active Directory domain.	MAJOR
8228	A connection to a directory server to obtain the subscription configuration cannot be obtained.	MAJOR

Log ID	Log message for Policy plug-in	Severity level
8229	The subscription configuration cannot be located. Requests cannot be handled for this plugin. For Active Directory, confirm that the configuration exists in any of the domains in the forest. For iPlanet and ADAM / AD LDS, confirm that the configuration exists in the domain in which subscription is running. Finally, confirm that the plugin user credentials are correct and that the user has permissions for obtaining the configuration.	MAJOR
8230	The subscription configuration could not be loaded. Requests cannot be handled for this plugin.	MAJOR
8235	The plugin is unable to obtain the current domain for the machine on which it is running.	MAJOR
8236	The root of the Active Directory forest could not be obtained.	MAJOR
8237	The current site for the plugin could not be obtained.	MAJOR
8238	A distinguished name was not provided from the client endpoint. Since the subscription configuration property marimba.subscriptionplugin.usednfromclientonly =true, the plugin will not attempt to resolve the distinguished name from only the user or machine name.	MAJOR
8239	Group membership cannot be determined for the user and machine distinguished names.	MAJOR
8240	The Subscription Plugin is unable to resolve user and machine distinguished names for the client endpoint.	MAJOR
8241	The Subscription Plugin is unable to obtain the containers to which the machine and user distinguished names belong.	MAJOR
8242	The initialization of the plugin failed. No requests can be handled.	MAJOR
8243	The property marimba.subscriptionplugin.requireentriesinldap =true. Subscription for ALL endpoints will not be sent.	MAJOR

Log ID	Log message for Policy plug-in	Severity level
8244	The credentials specified during the publish process are invalid. Please make sure that a username and password with the appropriate permissions has been specified.	MAJOR
8245	A connection could not be obtained for discovering an AD host. Please make sure that DNS is configured correctly for an Active Directory forest.	MAJOR
8246	The DNS format for the given domain in Netbios format cannot be obtained.	MAJOR
8247	Retry to initialize the Plugin, count:	CRITICAL
8248	Client request rejected because Plugin is initializing.	CRITICAL
8249	Invalid Active Directory dns domain format	MAJOR
8250	Only server or domain field can be specified for a Transmitter mapping.	MAJOR
8251	Unable to find user and machine information in directory - user, machine name, user domain, machine domain	MAJOR
8252	Unable to find user and machine information in directory - user, machine name	MAJOR
8253	The connection could not be established with the LDAP Server. Please verify your LDAP/Network settings.	MAJOR
8254	Invalid provisioning DN	MAJOR
8255	The Marimba configuration cannot be located. Requests cannot be handled for this plugin. For Active Directory, confirm that the configuration exists in any of the domains in the forest. For iPlanet and ADAM / AD LDS, confirm that the configuration exists in the domain in which Subscription is running. Finally, confirm that the plugin user credentials are correct and that the user has permissions for obtaining the configuration.	MAJOR
8256	Invalid pool size. Valid value should be a number between 1 and 100	MAJOR

Proxy

This section lists the log IDs, corresponding log messages, and log severity levels for the proxy.

Log ID	Log message for Proxy	Severity level
4154	IOException while processing request	MINOR
4200	Setting proxy root directory	AUDIT
4201	Starting proxy server on	AUDIT
4202	Stopping proxy server	AUDIT
4210	Setting up proxy as reverse proxy for	AUDIT
4211	Setting proxy in reverse secure mode	AUDIT
4212	Setting up proxy in normal mode	AUDIT
4213	Setting up proxy to chain to	AUDIT
4214	Removing proxy chain configuration	AUDIT
4220	Clearing the proxy cache	AUDIT
4221	Changing maximum cache size to	AUDIT
4222	Changing cache low water mark to	AUDIT
4230	Started cache collection at current cache size of	AUDIT
4231	Stopped cache collection at current cache size of	AUDIT
4232	No room in cache to insert file of length	AUDIT
4240	Refreshing cache with channel group	AUDIT
4241	Refreshing cache with car file	AUDIT
4250	Property added	AUDIT
4251	Property deleted	AUDIT
4252	Property modified	AUDIT
4300	Proxy is not processing any client-connections	AUDIT
4301	Proxy is processing client-connections	AUDIT
4302	Proxy is processing max number of client-connections	AUDIT

Log ID	Log message for Proxy	Severity level
4303	Proxy is processing less than the max number of client-connections	AUDIT
4304	Session migration disabled by	
4400	Exception occurred while starting up proxy channel	CRITICAL
4401	Error while loading license	CRITICAL
4402	Could not find evaluation license	CRITICAL
4403	Error opening proxy cache	CRITICAL
4404	Error writing to proxy cache	MAJOR
4405	Java 2 VM required, current VM version	CRITICAL
4406	Could not find a valid license to run the Proxy	CRITICAL
4410	Error while refreshing port properties	MAJOR
4411	Attempt to clear the proxy cache while proxy is running	WARNING
4412	Error with water mark specification	MINOR
4413	Error setting proxy root directory	MAJOR
4414	Could not resolve outgoing host address	MAJOR
4420	Invalid reverse proxy target url	MAJOR
4421	Error starting secure reverse proxy	MAJOR
4430	Exception while processing request from	MINOR
4431	Exception occurred while retrieving files from transmitter for URL	MINOR
4440	Could not open channel URL	MINOR
4441	Got invalid transmitter reply while updating channel	MINOR
4442	Exception while refreshing cache with URL	MINOR
4443	Exception while refreshing from car file	MINOR
4444	Cannot configure both proxy chain and reverse proxy at the same time	MAJOR
4445	Could not connect to proxy chain URL	MAJOR
4450	Unexpected HTTP reply from transmitter	MAJOR

Log ID	Log message for Proxy	Severity level
4451	No XML data received from transmitter	MAJOR
4452	Exception occurred with making an XML listing to the transmitter	MAJOR
4500	Sending status report	AUDIT
13312	Proxy was not shutdown properly last time, this might have caused storage corruptions. Run ProxyVerify to check for storage corruptions.	

Proxy Administrator

This section lists the log IDs, corresponding log messages, and log severity levels for Proxy Administrator.

Log ID	Log message for Proxy Administrator	Severity level
35700	Successfully connected to the proxy	AUDIT
35701	Failed to connect to the proxy	MAJOR
35702	Proxy admin in one to one mode	AUDIT
35703	Proxy admin in the one to N mode	AUDIT
35704	Proxy channel started	AUDIT
35705	Proxy channel stopped	AUDIT
35706	Preload proxy channel	AUDIT
35707	Proxy channel repair cache	AUDIT
35708	Proxy channel clear cache	AUDIT
35709	Proxy channel started on multiple endpoints	AUDIT
35710	Proxy channel stopped on multiple endpoints	AUDIT
35711	Preload multiple proxy channels	AUDIT
35712	Proxy channel repair cache on multiple endpoints	AUDIT
35713	Proxy channel clear cache on multiple endpoints	AUDIT
35714	Add url	AUDIT
35715	Cleaning all the proxy preload session attributes	AUDIT
35716	Preload proxy	AUDIT

Log ID	Log message for Proxy Administrator	Severity level
35717	Preload multiple proxies	AUDIT
35718	Proxy channel repair cache failed. Proxy server is running.	WARNING
35719	Proxy channel clear cache failed. Proxy server is running.	WARNING
35720	Publishing configuration for proxy	AUDIT
35721	Successfully changed the proxy root for proxy	AUDIT
35722	Failed to change the proxy root for proxy	AUDIT
35723	Proxy type for proxy	AUDIT
35724	Set SSL for proxy	AUDIT
35725	Set no SSL for proxy	AUDIT
35726	Administration access for proxy	AUDIT
35727	Admin log roll policy for proxy	AUDIT
35728	Access log roll policy for proxy	AUDIT
35729	Admin log roll versions for proxy	AUDIT
35730	Access log roll versions for proxy	AUDIT
35731	Failed to initialize the Tuner session	MAJOR
35732	Failed to initialize the Proxy admin	MAJOR
35733	Proxy Admin session timed out	MAJOR
35734	Internal Proxy Admin error	MAJOR

Report Center

This section lists the log IDs, corresponding log messages, and log severity levels for Report Center.

Log ID	Log message for Report Center	Severity level
29000	Report Center started	AUDIT
29001	Initialization failure. Start Report Center again	MINOR
29002	Using default database	AUDIT

Log ID	Log message for Report Center	Severity level
29003	There are no default reporting and inventory connections to the database. Contact your primary administrator	AUDIT
29004	There is no default reporting or inventory connections to the database. Contact your primary administrator	MAJOR
29005	Failed to initialize module. Start Report Center again	MINOR
29006	Failed to destroy module	MINOR
29007	There is no default inventory connection to the database. Contact your primary administrator	MAJOR
29008	Default reporting and inventory connections to the database are not pointing to the same server and SID. Contact your primary administrator	MAJOR
29009	Report Center failed to initialize access control list functionality. Start Report Center again	MAJOR
29010	Report Center requires JRE 1.4 or higher. Update the tuner to version 7.0.0.0 or higher because that version contains the required JRE	MAJOR
29011	Database's schema version is unsupported	MAJOR
29050	Failed to get plug-in configuration from the database	MINOR
29051	Received properties from the plug-in	AUDIT
29052	Using default plug-in configuration properties	AUDIT
29053	Received plug-in configuration properties from the database	AUDIT
29054	Using configuration properties from the plug-in	WARNING
29055	Failed to get configuration properties from the plug-in	MAJOR
29056	Invalid plugin URL	MAJOR
29057	Plug-in version should be 6.0 or higher	MAJOR
29058	Access control list functionality has been enabled	WARNING
29059	Access control list functionality has been revoked	WARNING

Log ID	Log message for Report Center	Severity level
29060	Extended privileges to standard administrators have been granted	WARNING
29061	Extended privileges to standard administrators have been revoked	WARNING
29062	Report Center debug flags are set	AUDIT
29100	Task Manager started	AUDIT
29101	Task Manager stopped	AUDIT
29102	Run schedule changed	AUDIT
29103	Scheduled queries will next run	AUDIT
29104	Task manager not started because collections are currently disabled for this report center. Another report center may currently be enabled to run collections	WARNING
29105	Task Manager cleaned up	AUDIT
29106	Could not connect to the database. Task Manager will retry connecting to the database at the next scheduled time.	WARNING
29107	Failed to clean up collection in LDAP	WARNING
29110	Default reporting connection to the database is not available	MAJOR
29111	Server is busy. Database connections are temporarily not available. Try again later	MAJOR
29112	Database has schema that is not supported	MAJOR
29113	An initialization error occurred while connecting to the database	MAJOR
29114	A database exception occurred	MAJOR
29115	Server is busy. Database connections are temporarily not available. Try again later	MAJOR
29116	Cannot use new database, reverting to old database	WARNING
29117	Database changed	AUDIT
29118	Database ACL User Manager is not initialized	MAJOR
29120	Task started	AUDIT

Log ID	Log message for Report Center	Severity level
29121	Task succeeded	AUDIT
29122	Task failed	MAJOR
29123	Task canceled	AUDIT
29124	Task pending	AUDIT
29130	Failed to create collection in LDAP	MAJOR
29131	Failed to add machines in LDAP	MAJOR
29132	Failed to commit collection in LDAP	MAJOR
29133	Collection does not contain machine names (machine_name column)	MAJOR
29134	The query name entered is too long for collections. WARNING Collection names can't exceed 64 characters	WARNING
29135	Task failed because collections are disabled	WARNING
29136	Report Center has been enabled to run collections	AUDIT
29137	Report Center has been disabled to run collections	AUDIT
29138	Collection does not contain machine names (machine_name column)	MAJOR
29139	Collection contains machine(s) with unknown domain	MAJOR
29140	LDAP is not configured	WARNING
29141	Could not connect to LDAP	MAJOR
29142	An LDAP exception occurred	MAJOR
29143	Subscription schema is unsupported or not installed in LDAP	MAJOR
29144	Collection contains duplicate machine names. Modify your query to return distinct machine names	MAJOR
29145	Marimba schema is not supported or not installed in LDAP	MAJOR
29160	Report Center cannot determine the current domain	MAJOR
29161	Deleted collection in LDAP	AUDIT
29162	Failed to delete collection in LDAP	AUDIT

Log ID	Log message for Report Center	Severity level
29163	The owner of the Collection must be a user in LDAP because Subscription access control lists are turned on	AUDIT
29164	Failed to create access control list for Collection owner in LDAP	AUDIT
29165	The machine was not added to the Collection group because the owner doesn't have the access control list write permission for the machine	AUDIT
29200	Created folder	AUDIT
29201	Deleted	AUDIT
29202	Created node	AUDIT
29203	Moved	AUDIT
29204	Permission denied	AUDIT
29205	Filtered state has been manually overwritten	WARNING
29206	Renamed	WARNING
29250	Failed to get transmitter port information	AUDIT
29251	Failed to get a session to	AUDIT
29252	Failed to get the root directory	AUDIT
29253	Failed to get the transmitter directory	AUDIT
29254	Failed to get the publish admin	AUDIT
29255	Transmitter URL	AUDIT
29256	Transmitter port info	AUDIT
29257	Failed to get the RPC publisher	MAJOR
29258	Failed to get the channel publisher	MAJOR
29259	Failed to get the publish index	MAJOR
29260	The size of the index is 0, so the channel most likely does not exist	MAJOR
29261	Committing publish	AUDIT
29262	Error while publishing	MAJOR
29300	Cannot get user information	WARNING

Log ID	Log message for Report Center	Severity level
29301	User denied access. Primary administrator access required	WARNING
29310	Plug-in url selected	AUDIT
29311	Received plug-in configuration	AUDIT
29312	Saved copy of original plug-in properties	AUDIT
29313	Plug-in properties displayed	AUDIT
29314	Error passing configuration settings between pages	WARNING
29315	Current property changes applied	AUDIT
29316	Endpoint properties displayed	AUDIT
29317	Cancelled plug-in configuration changes	AUDIT
29318	Preview plug-in configuration changes	AUDIT
29319	Cancel preview plug-in configuration	AUDIT
29320	Added new property	AUDIT
29321	Changed property value	AUDIT
29322	Deleted property	AUDIT
29323	Successfully published configuration	AUDIT
29324	Failed to publish configuration	AUDIT
29350	Failed to list custom views from the database	MAJOR
29360	Missing query library folder in the default profile	MAJOR
29365	Failed to query machine detail component	WARNING
29370	Setting schedule for item	AUDIT
29371	Successfully set a schedule for item	AUDIT
29372	The scheduled form query does not have a default query value	AUDIT
29373	SMTP host has not been set up in System Settings	WARNING
29374	Removed a scheduled item	AUDIT
29375	Successfully executed scheduled query	AUDIT
29376	Failed to execute scheduled query	WARNING
29377	Failed to get query manager context	MAJOR

Log ID	Log message for Report Center	Severity level
29378	A SQL exception has occurred	WARNING
29379	Triggered scheduled query	AUDIT
29380	Failed to trigger scheduled query	WARNING
29381	Added schedule	AUDIT
29382	Edited schedule	AUDIT
29383	Access control list exception occurred. Query can not be executed for the user	WARNING
29384	Report can not be found	WARNING
29385	Folder does not contain any queries to run	WARNING
29400	Emergency administrator is not allowed to log in because Report Center access control list functionality is turned on	MAJOR
29401	User must be authenticated using directory service because Report Center access control list functionality is turned on	MAJOR
29402	Failed to register user with Report Center ACL Manager	MAJOR
29403	Registered user with Report Center ACL Manager	AUDIT
29404	Failed to synchronize user related LDAP information to database	MAJOR
29405	CMS LDAP to database synchronization task is not available or failed to initialize properly. Check CMS logs for more information	MAJOR
29406	Unregistered user with Report Center ACL Manager	AUDIT
29407	Failed to unregister user with Report Center ACL Manager	WARNING

Schema Management

This section lists the log IDs, corresponding log messages, and log severity levels for Schema Management.

Log ID	Log message for Schema Management	Severity level
40000	Loading script - package	AUDIT
40001	Error loading script - package	MAJOR
40002	Duplicate task	WARNING
40020	Set database to	AUDIT
40021	Failed to connect to database on host	MAJOR
40022	Disconnecting	AUDIT
40023	No valid database connection was found	MAJOR
40030	Download script	AUDIT
40031	Failed to download script	MAJOR
40040	Execute script	AUDIT
40041	Using script - environment	AUDIT
40042	Failed to execute script	MAJOR
40043	Finished executing script	AUDIT
40050	Failed executing script command	WARNING
40051	Script error	MAJOR
40060	Install the Query Library	AUDIT
40061	Failed to install the Query Library	MAJOR
40062	Cannot install the Query Library	AUDIT
40063	Installed the Query Library	AUDIT
40070	Terminating logged on users	AUDIT
40071	Error terminating users	MINOR
40081	Failed to connect to LDAP on host:	MAJOR
40082	Invalid Collection Base	MAJOR
40083	Invalid schema	MAJOR

Log ID	Log message for Schema Management	Severity level
40091	Invalid schema version. Check your schema version.	MAJOR
40092	Failed to modify ACL objects. Check your credentials and permissions on the ACL container.	MAJOR
40093	No all_all objects exist under the container.	MAJOR
40094	Changing the following ACL entry:	AUDIT
40095	Migrate domain policies	MAJOR
40096	Delete child containers	MAJOR
40097	Object not found	MAJOR
40100	Unknown command-line argument	MAJOR
40101	Schema Manager: Command line successfully executed	AUDIT
40102	Database connection succeeded	AUDIT
40103	Database schema installed	AUDIT
40104	Install the Core schema	AUDIT
40105	Database schema uninstalled	AUDIT
40106	LDAP schema generation failed	MAJOR
40107	LDAP schema generated	MAJOR
40108	Schema Manager: Command line failed	MAJOR
40109	Invalid schema name:	MAJOR
40110	Failed to initialize the command line	MAJOR
40130	CMS task creation succeeded	AUDIT
40131	CMS task creation failed	AUDIT
40132	CMS task removal succeeded	AUDIT
40133	CMS task removal failed	AUDIT
40140	Successfully exported the dbtree	AUDIT
40141	Dbtree export failed	MAJOR
40142	Successfully imported the dbtree	AUDIT
40143	Dbtree import failed	MAJOR
40144	Dbtree table is not empty	MAJOR

Log ID	Log message for Schema Management	Severity level
40145	Failed to create file to export dbtree.	MAJOR
40146	Exported dbtrees successfully	
40147	Imported dbtrees successfully	

Setup and Deployment

This section lists the log IDs, corresponding log messages, and log severity levels for the Setup and Deployment components.

Log ID	Log message for Setup and Deployment	Severity level
36000	Copy segment started	
36001	Copy segment ended	
36002	Copy failed	
36003	Copy cancelled	
36004	Copy done:	
36005	Copy starting	
36006	Copy partially cancelled	
36007	Security unavailable	
36008	Copy exception:	
36009	Uploading of CAR file started	
36010	Uploading of CAR file ended	
36011	Uploading of URL channel started	
36012	Uploading of URL channel ended	
36013	Invalid channel URL	
36014	Creation of CAR file started	
36015	Creation of CAR file ended	
36016	Deletion of CAR file started	
36017	Deletion of CAR file ended	
36018	Editing of CAR file started	
36019	Editing of CAR file ended	

Log ID	Log message for Setup and Deployment	Severity level
36020	Remapping of CAR files started	
36021	Remapping of CAR files ended	
36022	No matching segments	
36023	Failed to remap CAR file	
36024	CAR file does not exist	
36500	DeployWizard module successfully started	AUDIT
36520	Missing license source type	MAJOR
36521	Invalid license source type	MAJOR
36522	Missing license key	MAJOR
36523	Missing username and/or password	MAJOR
36524	Missing license file location	MAJOR
36525	Invalid license	MAJOR
36526	Invalid license file location	MAJOR
36527	Cannot retrieve license file	MAJOR
36528	Failed license installation	MAJOR
36540	Cannot retrieve channel list	MAJOR
36541	Product list URL is malformed	MAJOR
36542	Product list file from URL cannot be found or has bad format	MAJOR
36543	Cannot find product list file on CD	MAJOR
36544	Product list file from CD has bad format	MAJOR
36545	Cannot find product list file in channel storage	MAJOR
36546	Product list file from channel storage has bad format	MAJOR
36547	No CAR files found on the given location	MINOR
36560	Missing install source type	MAJOR
36561	Invalid install source type	MAJOR
36562	Missing install source location	MAJOR
36563	Install cd location is not a directory	MAJOR

Log ID	Log message for Setup and Deployment	Severity level
36564	Bad install source URL	MINOR
36580	Missing transmitter host and/or administration port number	MAJOR
36581	Invalid transmitter administration port number	MAJOR
36582	Missing transmitter listen port	MAJOR
36583	Invalid transmitter listener port number	MAJOR
36584	Failed to start transmitter	MAJOR
36585	Failed to get transmitter's launcher, workspace, tunerconfig, and/or services	MAJOR
36586	Missing transmitter channel	MAJOR
36587	Failed to stop transmitter	MAJOR
36588	Failed to rename transmitter URL	MAJOR
36589	Timeout waiting for transmitter to start	MAJOR
36600	Missing copy channel source	MAJOR
36601	Failed to copy channels	MAJOR
36602	Channel copy failed at source	MAJOR
36603	Channel copy failed at destination	MAJOR
36604	Copy source missing index	MAJOR
36605	Copy source has broken signature	MAJOR
36606	Missing required files at destination	MAJOR
36607	Successfully copied channel	AUDIT
36608	Failed to publish channel to destination	MAJOR
36609	Source access denied	MAJOR
36610	Destination access denied	MAJOR
36620	Workflow cleanup successfully completed	AUDIT
36621	User cancelled installing products	AUDIT
36640	Failed to update channels	MAJOR
36641	Failed to change update URL, waiting for channel restart	AUDIT
36642	Successfully changed update URL	AUDIT

Log ID	Log message for Setup and Deployment	Severity level
36643	Successfully changed Primary channel URL	AUDIT
36644	Successfully subscribed channel	AUDIT
36645	Failed to subscribe channel	MINOR
36646	Successfully started newly-subscribed channel	AUDIT
36647	Failed to start newly-subscribed channel	MINOR
36648	Failed to get remote transmitter's launcher, workspace, and/or tunerconfig	MAJOR
36660	Bad URL in source products base list	MAJOR
36661	Unauthorized to access source transmitter	MAJOR
36662	Failed to connect to source transmitter	MAJOR
36663	Connect to source transmitter status	AUDIT
36664	Unknown source transmitter host	MAJOR
36665	Bad source transmitter XML listing format	MAJOR
36666	Source transmitter I/O error	MAJOR
36667	Cannot locate CAR file(s)	MAJOR
37001	Deleting the Profile	AUDIT
37002	Deleting the Profile	AUDIT
37003	Successfully created the Profile	AUDIT
37004	Failed to create the Profile	MINOR
37005	Updating Profile	AUDIT
37006	Successfully updated the Profile	AUDIT
37007	Failed to update the Profile	MINOR
37008	Successfully deleted the Profile	AUDIT
37009	Failed to delete the Profile	MINOR
37500	Starting to copy CAR files	
37501	Starting to copy script files	
37502	Starting to create installer	
37503	Starting to compress dictionary	
37504	Starting to modify stub package args	

Log ID	Log message for Setup and Deployment	Severity level
37505	Starting to compress zip	
37506	Starting to modify stub package	
37507	Starting to write setup.ini file	
37508	Copying source files to destination succeeded	
37509	Copying template files to destination succeeded	
37510	Modifying msi db file succeeded	
37511	Modifying setup.ini file succeeded	
37512	Copying CAR files succeeded	
37513	Copying script files succeeded	
37514	Creating bin directory succeeded	
37515	Copying source files to destination failed	
37516	Copying template files to destination failed	
37517	Modifying msi db file failed	
37518	Modifying setup.ini file failed	
37519	Failed to copy CAR file	
37520	Failed to append file	
37521	Failed to copy script file	
37522	Failed to make installer	
37523	Failed to write file	
37524	Failed to compress directory	
37525	Failed to compress zip	
37526	Failed to write setup.ini file	
37527	Failed to create bin directory	
37528	Failed to create installer	
37529	Failed due to incompatible template	
37530	Modifying installer properties file failed	

Storage

This section lists the log IDs, corresponding log messages, and log severity levels for the transmitter and tuner storage.

Log ID	Log message for Storage	Severity level
13000	Compact started	INFO
13001	Compact finished	INFO
13100	Storage opened	AUDIT
13101	Storage closed	AUDIT
13102	Storage upgrade started	AUDIT
13103	Storage upgrade finished	AUDIT
13200	Error during compaction	MAJOR
13201	Inode table encountered deleted inode	MAJOR
13203	Storage inode table missing	MAJOR
13204	Unexpected I/O error	MAJOR
13300	DB file header format error	CRITICAL
13301	DB file version mismatch	CRITICAL
13302	Inode table header format error	CRITICAL
13304	Inode table version mismatch	CRITICAL
13305	Unable to repair storage	CRITICAL
13306	Closing leaked IFile	WARNING
13307	Compaction disabled	INFO
13308	Compaction enabled	INFO
13309	Copy compaction failed	WARNING
13310	Couldn't rename file	WARNING
13311	Aborting compaction	WARNING

Subnet Repeater Policy

This section lists the log IDs, corresponding log messages, and log severity levels for the Subnet Repeater Policy:

Log ID	Log message for Subnet Repeater Policy	Severity level
34000	Allow arbitrary	AUDIT
34001	Subnet repeater policy initialized	AUDIT
34002	Default ordering	AUDIT
34003	Unknown ordering	WARNING
34004	Subnet repeater policy started	AUDIT
34005	Number of networks loaded	AUDIT
34006	Network	AUDIT
34007	Default network	AUDIT
34008	Only one default network supported	WARNING
34009	Malformed URL	WARNING
34010	Syntax error, found a mismatched network tag	WARNING
34011	Found and ignoring empty network	WARNING
34012	Configuration is missing DOCTYPE and cannot be validated	WARNING
34013	Configuration is invalid	MAJOR
34014	Configuration error	MAJOR
34015	XML Parser error	MAJOR
34016	Found	AUDIT
34017	No config.xml file present in subnet repeater policy extension	CRITICAL

Transmitter

This section lists the log IDs and corresponding log messages for the transmitter.

Log ID	Log message for Transmitter	Severity level
2200	Channel created	—
2201	Channel updated	AUDIT
2202	Channel deleted	—
2203	Failed to delete channel	MAJOR
2204	Admin logon	—
2205	Admin logoff	—
2206	Admin config change	—
2207	Transmitter running with demo license	—
2208	Transmitter Module Started	AUDIT
2209	Transmitter Module Stopped	AUDIT
2210	Publish Module Started	AUDIT
2211	Publish Module Stopped	AUDIT
2212	Replication Module Started	AUDIT
2213	Replication Module Stopped	AUDIT
2214	Error initializing tuner administration port	CRITICAL
2215	Incorrect client cert for publishing channel	AUDIT
2216	Access forbidden for publishing channel	AUDIT
2217	Incorrect password for publishing channel	AUDIT
2218	Transmitter node deleted from workspace object	—
2219	Low on disk space	WARNING
2220	Transmitter HTTP port changed	AUDIT
2221	Transmitter title changed	AUDIT
2222	Logs directory changed	AUDIT
2223	Log settings changed	AUDIT
2224	Bind address changed	AUDIT

Log ID	Log message for Transmitter	Severity level
2225	Local client connections only changed	AUDIT
2226	Trusted publish hosts changed	AUDIT
2227	Trusted publish netmask changed	AUDIT
2228	Publish email changed	AUDIT
2229	Publish SMTP server changed	AUDIT
2230	Publish email notification always changed	AUDIT
2231	Redirect to active repeaters changed	AUDIT
2232	Repeater redirection strategy changed	AUDIT
2233	Master participation in redirection changed	AUDIT
2234	Client repeater timeout changed	AUDIT
2235	Down repeater timeout changed	—
2236	List of master transmitters to be repeated changed	AUDIT
2237	Replication interval changed	AUDIT
2238	Mirroring enabled changed	AUDIT
2239	Mirroring host changed	AUDIT
2240	Mirroring permissions changed	AUDIT
2241	Maximum number of concurrent connections changed	AUDIT
2242	Index cache settings changed	—
2243	Disk cache settings changed	—
2244	Compression enabled changed	AUDIT
2245	Compression settings changed	AUDIT
2246	HTTP timeout changed	AUDIT
2247	Bandwidth limit changed	AUDIT
2248	Authentication Method: LDAP	AUDIT
2249	Authentication Method: Custom Authenticator	AUDIT
2250	Authentication Method: Local User Database	AUDIT
2251	Admin authentication settings changed	AUDIT
2252	Admin emergency password changed	AUDIT

Log ID	Log message for Transmitter	Severity level
2253	SSL certificate not found	CRITICAL
2254	SSL password incorrect	CRITICAL
2255	SSL enabled	AUDIT
2256	SSL disabled	AUDIT
2257	SSL client certificates changed	—
2258	SSL strong encryption changed	AUDIT
2259	Inherit SSL settings from main port changed	AUDIT
2260	The certificate cannot be used with the export security library as its RSA modulus is greater than 512 bits	CRITICAL
2261	Byte-level differencing enabled changed	AUDIT
2262	Byte-level settings changed	—
2263	Channel segment deleted	—
2264	Directory node deleted from transmitter workspace object	—
2265	Directory node created in transmitter workspace object	—
2266	Channel node moved in transmitter workspace object	—
2267	Directory node moved in transmitter workspace object	—
2268	Shutting down existing connection	—
2269	Credentials required for publishing channel	AUDIT
2270	Subscribe permission changed for transmitter workspace object	—
2271	Publish permission changed for transmitter workspace object	—
2272	Replication permission changed for transmitter workspace object	—
2273	Subscribe permission changed for directory workspace object	—

Log ID	Log message for Transmitter	Severity level
2274	Publish permission changed for directory workspace object	—
2275	Replication permission changed for directory workspace object	—
2276	Subscribe permission changed for channel workspace object	—
2277	Publish permission changed for channel workspace object	—
2278	Replication permission changed for channel workspace object	—
2279	Transmitter workspace object hidden	—
2280	Transmitter workspace object unhidden	—
2281	Directory workspace object hidden	—
2282	Directory workspace object unhidden	—
2283	Channel workspace object hidden	—
2284	Channel workspace object unhidden	—
2285	Amount of memory provided for diffs changed for this transmitter to	AUDIT
2286	Amount of the minimum free disk space percentage changed for this transmitter to	AUDIT
2287	Down repeater timeout intervals changed	AUDIT
2288	Repeater update interval changed	AUDIT
2289	DSL integration initialized successfully	AUDIT
2290	DSL integration failed to initialize, check properties	MAJOR
2291	Stopping Transmitter	AUDIT
2292	Repeater sync blackout schedule changed	
2293	Tuner session migration disabled by	
2294	Segmentation hierarchy of transmitter changed to	AUDIT
2295	Custom Segmentation URL of transmitter changed to	AUDIT
2300	User limit exceeded	—

Log ID	Log message for Transmitter	Severity level
2301	Authentication error	—
2302	Update SDK client denied	—
2303	Invalid transmitter license	CRITICAL
2304	Channel requires SSL	—
2305	Plugin error occurred	WARNING
2306	Replication slave error	MAJOR
2307	HTTP Listener threadpool limit exceeded	MAJOR
2308	License about to expire in the following number of days	—
2309	License expired	—
2310	User limit above threshold	MAJOR
2311	Unexpected exception occurred	CRITICAL MAJOR WARNING
2312	Unable to bind socket (Bind exception occurred)	CRITICAL
2313	Network error occurred	INFO
2314	Internal transmitter error	CRITICAL
2315	Recreated missing transmitter workspace object	MAJOR
2316	LDAP error occurred	MAJOR
2317	Publish operation timed out	WARNING
2318	File not found	MINOR
2319	Java 2 VM required, current VM version	CRITICAL
2320	Exception starting segment	—
2321	Illegal Transmitter HTTP provided. The HTTP port value should be an integer between 0 and 65535	CRITICAL
2322	Could not find a valid license to run the Transmitter	CRITICAL
2323	IO Error	MAJOR
2324	DSL integration could not connect to CMS	MINOR

Log ID	Log message for Transmitter	Severity level
2325	DSL integration failed to activate published channel	MINOR
2326	DSL integration failed to inactivate deleted channel	MINOR
2327	Failed to open file storage	CRITICAL
2328	Failed to open index storage	CRITICAL
2401	Replication: repeater created channel	—
2402	Replication: repeater updated channel	—
2403	Replication: master transmitter license expired	—
2404	Replication: master transmitter user limited exceeded	—
2405	Replication: master transmitter busy	—
2406	Replication: error occurred on master	—
2407	Replication: No such channel on replication master	—
2408	Replication: access denied by master	—
2409	Replication: protocol error	—
2410	Replication: protocol unsupported by master	—
2411	Cannot connect to RPC listener on master transmitter	—
2412	No transmitter service on master transmitter	—
2413	Incorrect administrator password for mirror master	—
2414	Cannot determine transmitter's RPC port for mirroring	—
2415	Unexpected exception while replicating	—
2416	Cannot connect to master	—
2417	Scheduled repeater update skipped (due to tuner schedule restrictions) until	INFO
2418	Repeater updates disabled due to tuner schedule restrictions	WARNING

Log ID	Log message for Transmitter	Severity level
2419	Repeater removed (timed-out) due to not checking in for updates	AUDIT
2420	Deleting	AUDIT
2421	Too many network errors - giving up replication for now	MAJOR
2422	Repeater added to list of known repeaters	AUDIT
2423	Invalid repeater	MINOR
2424	Scheduled repeater update skipped (due to blackout period)	—
2425	Error parsing repeater sync blackout schedule property	
2426	Tuner upgrade to 8.0 required to be able to use repeater sync blackout schedule	
2501	Not acting as a repeater for any transmitters	AUDIT
2502	Not running as a mirror slave	AUDIT
2503	Repeating from master	AUDIT
2504	Mirroring from master	AUDIT
2505	Repeater deleting master entry, that it should no longer use for replication	—
2506	Listing request status	AUDIT
2507	File cache status	—
2508	Index cache status	—
2509	Successful replication completion status	—
2510	Segment index loaded	—
2511	Segment indexes unloaded	—
2512	Failed replication completion status	—
2513	Listing request started	AUDIT
2514	Listing request completed	AUDIT
2515	Listing request failed	AUDIT
2601	Upgrading workspace	—
2602	Upgrading transmitter	AUDIT

Log ID	Log message for Transmitter	Severity level
2603	Upgrading channel	AUDIT
2604	Upgrading segment	AUDIT
2605	Workspace upgrade complete	AUDIT
2610	Upgrading workspace tree structure	—
2611	Workspace tree structure upgraded	—
2612	Upgrading transmitter properties	—
2613	Transmitter Properties upgraded	—
2614	Upgrading indexes	—
2615	Indexes upgraded	—
2616	Found v46 index	—
2617	Upgrading files	—
2618	Files upgraded	—
2619	Found partial file	—
2620	Found zero-length compressed file	—
2650	Specified transmitter root is not a directory	—
2651	Specified directory is not a transmitter directory	—
2652	Transmitter version cannot be upgraded	—
2653	Unexpected exception during upgrade	CRITICAL MAJOR
2654	Unexpected directory in workspace	MAJOR
2655	Failed to open storage	—
2656	Failed to upgrade segment	—
2657	Failed to delete v46 index	—
2658	Failed to read index	—
2700	Transmitter is not processing any client-connections	AUDIT
2701	Transmitter is processing client-connections	AUDIT
2702	Transmitter is processing max number of client-connections	AUDIT

Log ID	Log message for Transmitter	Severity level
2703	Transmitter is processing less than the max number of client-connections	AUDIT
2750	Sending status report	AUDIT
2775	No user specified for email notification	
2776	SMTP port is not specified, using the default port	
2777	Invalid SMTP port is specified, using the default port	

Transmitter Administrator

This section lists the log IDs, corresponding log messages, and log severity levels for Transmitter Administrator.

Log ID	Log message for Transmitter Administrator	Severity level
35400	Connected to Transmitter	AUDIT
35401	Failed to connect to Transmitter	MAJOR
35402	Started Transmitter	AUDIT
35403	Stopped Transmitter	AUDIT
35404	Synchronized Transmitter contents	AUDIT
35405	Applying 1-to-N action	AUDIT
35406	Existing 1-to-N action in progress	WARNING
35407	Clear file cache	AUDIT
35408	Clear index cache	AUDIT
35409	Publish Transmitter settings	AUDIT
35410	Applying delete Transmitter node	AUDIT
35411	Deleted Transmitter channel	AUDIT
35412	Deleted Transmitter segment	AUDIT
35413	Deleted Transmitter folder	AUDIT
35414	Applying move Transmitter node	AUDIT
35415	Renamed Transmitter channel	AUDIT

Log ID	Log message for Transmitter Administrator	Severity level
35416	Failed to rename Transmitter channel	WARNING
35417	Applying add Transmitter node	AUDIT
35418	Added Transmitter channel	AUDIT
35419	Failed to add Transmitter channel	WARNING
35420	Applying download car file	AUDIT
35421	Downloaded car file	AUDIT
35422	Failed to download car file	WARNING
35423	Show Transmitter channel	AUDIT
35424	Hide Transmitter channel	AUDIT
35425	Change Transmitter channel permissions	AUDIT
35426	Change Transmitter folder permissions	AUDIT
35427	Creating folder	AUDIT
35428	Created folder	AUDIT
35429	Failed to create folder	WARNING
35430	Applying delete user action	AUDIT
35431	Deleted user	AUDIT
35432	Failed to delete user	WARNING
35433	Applying add user action	AUDIT
35434	Failed to add user	WARNING
35435	Failed to add user to group	WARNING
35436	Applying add group action	AUDIT
35437	Failed to add group	WARNING
35438	Applying edit group action	AUDIT
35439	Applying delete group action	AUDIT
35440	Deleted group	AUDIT
35441	Failed to delete group	WARNING
35442	Applying edit user action	AUDIT
35443	Failed to edit user	WARNING
35444	Failed to retrieve node information	WARNING

Log ID	Log message for Transmitter Administrator	Severity level
35445	Folder already exists	WARNING
35450	Transmitter Admin session timed out	MAJOR
35451	Failed to initialize the Tuner session	MAJOR
35452	Failed to initialize the Transmitter admin	MAJOR
35453	Internal Transmitter Admin error	MAJOR

Tuner

This section lists the log IDs, corresponding log messages, and log severity levels for the tuner.

Log ID	Log message for Tuner	Severity level
1000	Kernel started	AUDIT
1001	Kernel arguments received	AUDIT
1002	Scheduler started	INFO
1003	Kernel RPC lock opened	AUDIT
1004	Kernel RPC lock attempted	AUDIT
1005	Kernel interactor changed	AUDIT
1006	Kernel property changed	INFO
1007	Kernel exited	AUDIT
1008	Kernel warning	MINOR
1010	Runchannel exited	AUDIT
1011	Kernel entering minimal mode	AUDIT
1012	Kernel exited minimal mode to show about dialog	AUDIT
1013	Kernel exited minimal mode to handle RPC connection	AUDIT
1014	Kernel exited minimal mode to handle DDE request	AUDIT
1015	Kernel exited minimal mode to launch primary channel	AUDIT
1016	Kernel exited minimal mode to start channel	AUDIT

Log ID	Log message for Tuner	Severity level
1017	Kernel exited minimal mode to update channel	AUDIT
1018	Kernel authenticator started	AUDIT
1019	Kernel authenticator error	MAJOR
1020	Kernel authenticator policy has wrong format	MAJOR
1021	Kernel authenticator stopped	AUDIT
1022	Kernel exited minimal mode to run Reschedule Event	AUDIT
1023	Bad global schedule	WARNING
1025	Kernel Car File Installer successfully installed .car channel	AUDIT
1026	Kernel Car File Installer started	AUDIT
1027	Kernel Car File Installer finished	AUDIT
1028	Kernel Car File Installer failed to install .car channel	MINOR
1029	Kernel Car File Installer started installing .car channel	AUDIT
1030	Kernel Car File Installer receive invalid arguments	MAJOR
1031	Kernel Car File Installer failed	MAJOR
1032	Kernel register signal handler failed	MAJOR
1033	Kernel exited minimal mode due to network availability	AUDIT
1034	Kernel exited minimal mode due to user logon	AUDIT
1035	Kernel exited minimal mode due to user logoff	AUDIT
1050	Workspace error	MAJOR
1051	Workspace damaged channel found	MAJOR
1052	Workspace flushed	INFO
1053	Workspace id generated	INFO
1054	Workspace filemap error	INFO
1100	Channel created	AUDIT
1101	Channel update started	AUDIT

Log ID	Log message for Tuner	Severity level
1102	Channel update finished	AUDIT
1103	Could not connect to host	MAJOR
1104	Channel update canceled	AUDIT
1105	Channel update auth required	AUDIT
1106	Channel update proxy bypassed	MINOR
1107	Channel index installed	AUDIT
1108	Channel unsubscribed	AUDIT
1109	Channel removed	AUDIT
1110	Channel checkpoint created	AUDIT
1111	Channel reverted to checkpoint	AUDIT
1112	Channel checkpoint deleted	AUDIT
1113	Channel URL changed	AUDIT
1114	Channel IP address updated	AUDIT
1115	Channel repeater list received	AUDIT
1116	Channel repeater marked down	AUDIT
1118	Channel update scheduled	INFO
1119	Channel start scheduled	INFO
1120	File map updated	AUDIT
1121	File map error	MAJOR
1122	Undo checkpoint failed	WARNING
1123	Channel verified	AUDIT
1150	Channel instance started	AUDIT
1151	Channel instance arguments received	AUDIT
1152	Channel instance stopped	AUDIT
1154	Channel instance created service	INFO
1155	Channel instance connecting to plugin	INFO
1200	Kernel update started	AUDIT
1201	Kernel update downloading	AUDIT
1202	Kernel update available	AUDIT

Log ID	Log message for Tuner	Severity level
1203	Kernel update no changes	AUDIT
1204	Kernel update finished	AUDIT
1205	Kernel update canceled	AUDIT
1206	Kernel update error	MAJOR
1250	Kernel restart wait time	AUDIT
1251	Kernel restart waiting on	AUDIT
1252	Kernel restart wait expired	WARNING
1400	RPC listener started	AUDIT
1401	RPC listener stopped	AUDIT
1402	RPC ssl listener started	AUDIT
1403	RPC ssl listener stopped	AUDIT
1404	RPC ssl set context	AUDIT
1405	RPC listener socket bound on	MAJOR
1406	RPC listener internal warning	WARNING
1407	RPC listener internal error	MAJOR
1408	RPC ssl listener error	MAJOR
1450	RPC session started	AUDIT
1451	RPC session stopped	AUDIT
1452	RPC ssl session started	AUDIT
1453	RPC ssl session stopped	AUDIT
1454	RPC session error	MAJOR
1455	RPC session opened stream	INFO
1456	RPC session closed stream	INFO
1457	RPC session stream error	MAJOR
1458	RPC session security test failed	MAJOR
1459	RPC session timeout	MAJOR
1500	RPC connection to remote machine started	AUDIT
1501	RPC connection to remote machine stopped	AUDIT
1502	RPC ssl connection to remote machine started	AUDIT

Log ID	Log message for Tuner	Severity level
1503	RPC ssl connection to remote machine stopped	AUDIT
1504	RPC connection to remote machine will be retried	AUDIT
1505	RPC connection to remote machine error	MAJOR
1506	RPC connection to remote machine opened stream	INFO
1507	RPC connection to remote machine closed stream	INFO
1508	RPC connection to remote machine has a stream IO error	MAJOR
1509	RPC connection to remote machine has an internal stream error	MAJOR
1510	RPC connection to remote machine could not be established	MAJOR
1511	RPC connection to remote machine encountered HTTP Proxy error	MAJOR
1600	License Manager has been initialized	AUDIT
1601	Trying to get license for	AUDIT
1602	Error getting license	MINOR
1603	Installed license for	AUDIT
1604	Error installing license	MAJOR
1605	Removed license for	AUDIT
1606	Error removing license	MAJOR
1701	SSL Certificate matched only with domain name	AUDIT
1702	Certificate not trusted for use	MAJOR
1703	Certificate expired	MAJOR
1704	Certificate name mismatch	MAJOR
1705	Unknown certificate issuer	MAJOR
1706	Certificate file missing	MAJOR
1707	Invalid certificate	MAJOR
1708	Invalid certificate signature	MAJOR
1709	Signature file missing	MAJOR

Log ID	Log message for Tuner	Severity level
1710	New certificate added	AUDIT
1711	Certificate use modified	AUDIT
1712	Certificate imported	AUDIT
1713	Certificate import failed	MINOR
1714	Certificate database saved	AUDIT
1715	Certificate database save error	MAJOR
1716	Certificate deleted	AUDIT
1717	Certificate database format error	CRITICAL
1718	Certificate issuers updated	AUDIT
1719	Certificate request completed	AUDIT
1720	Certificate request install failed	MINOR
1721	Certificate password check	AUDIT
1722	Certificate password changed	AUDIT
1723	Signing manifest parsing error	MAJOR
1724	Signing manifest missing	MAJOR
1725	Incorrect channel signature	MAJOR
1726	Certificate error overridden	AUDIT
1727	Certificate database upgraded	AUDIT
1728	Certificate database upgrade failed	CRITICAL
1730	SSL client certificate requested	AUDIT
1731	SSL client certificate sent	AUDIT
1732	SSL connection error	MAJOR
1733	SSL cipher suite changed	AUDIT
1734	SSL certificate chain incomplete	MAJOR
1735	SSL invalid certificate	MAJOR
1737	SSL invalid certificate chain	MAJOR
1739	SSL invalid certificate	MAJOR
1741	SSL error setting private key	MAJOR
1742	SSL exception during read	MAJOR

Log ID	Log message for Tuner	Severity level
1743	SSL exception during write	MAJOR
1744	SSL no client certificate presented	MAJOR
1745	Security exception	MAJOR
1746	Desktop init failed	MAJOR
1747	Desktop duplication icon id	MINOR
1748	Desktop too many icon id requests	MINOR
1749	Desktop add shortcut failed	MINOR
1750	Desktop add task icon failed	MINOR
1751	Desktop register service failed	MINOR
1752	Desktop unregister service failed	MINOR
1753	Desktop send to service failed	MINOR
1754	Desktop no dialup available	MINOR
1755	Desktop dialup server changed	AUDIT
1756	Desktop dialup idle timeout changed	AUDIT
1757	Desktop dialup user config	AUDIT
1758	Desktop dialup user config failed	MINOR
1759	Desktop dialup no user found	MINOR
1760	Desktop dialup no password found	AUDIT
1761	Desktop dialup connecting	AUDIT
1762	Desktop dialup connect failed	MAJOR
1763	Desktop dialup connected	AUDIT
1764	Desktop dialup disconnected	AUDIT
1766	Desktop create hidden console failed	MAJOR
1767	System class loader is not allowed to load classes in this package. For Marimba channels, these classes do not exist and are not necessary. For custom channels, check that you have included these classes in the channel	AUDIT
1768	Tuner anonymized successfully	AUDIT
1769	Tuner could not be anonymized	AUDIT

Log ID	Log message for Tuner	Severity level
1770	Tuner anonymizer supported only in Windows Platform	MAJOR
1771	Desktop create terminator failed	MAJOR
1772	Tuner anonymization canceled	AUDIT
1775	Machine wake up occurred, type of wake-up was WOW	INFO
1776	Machine wake up occurred, type of wake-up was Unknown	INFO
1777	Tuner no longer observing wake-up notification	INFO
1780	Network detection policy created	AUDIT
1781	Network detection policy changed	AUDIT
1782	Network detection policy unknown	MAJOR
1783	Network detection policy error	MAJOR
1784	Network online	AUDIT
1785	Network offline	AUDIT
1786	Network online delay changed	AUDIT
1787	Network offline delay changed	AUDIT
1788	Network plug initialization failed	MAJOR
1850	Token provider	AUDIT
1851	Token provider disabled	AUDIT
1852	Token provider anonymized	AUDIT
1853	Token provider exception	MAJOR
1854	Token provider generated key	AUDIT
1900	Error creating receipt storage	MAJOR
1901	Error reading back receipt from storage	MAJOR
1902	Error creating index	MAJOR
1903	Error expiring receipts	MAJOR
1904	Error deleting file	MAJOR
1905	Error creating receipt directory	MAJOR
1906	Error creating receipt index directory	MAJOR

Log ID	Log message for Tuner	Severity level
1907	Error purging receipts during scheduled event	MAJOR
1908	Error deleting index	MAJOR
1909	Receipt saved	AUDIT
1910	Receipt deleted	AUDIT
1985	Lite Weight Tuner Admin Logs	INFO
49118	Channel Peer Phase of MESH initiated	INFO
49119	Currently in File Peer Phase of MESH	INFO
49120	Skipping File Peer Phase in MESH	INFO
49121	All peers visited at end of ADP request	INFO
49122	File Peer Phase: Full mode peers identified	INFO
49123	File Peer Phase: Partial mode peers identified	INFO
49124	File Peer Phase: Minimal mode peers identified	INFO

Tuner Administrator

This section lists the log IDs, corresponding log messages, and log severity levels for Tuner Administrator.

Log ID	Log message for Tuner Administrator	Severity level
35100	Failed to connect to endpoint	MAJOR
35101	Successfully connected to endpoint	AUDIT
35102	Updating Inventory Service on endpoint	AUDIT
35103	Failed to update Inventory Service on endpoint	MINOR
35104	Successfully updated Inventory Service on endpoint	AUDIT
35105	Updating Subscription Service on endpoint	AUDIT
35106	Failed to update Subscription Service on endpoint	MINOR
35107	Successfully updated Subscription Service on endpoint	AUDIT
35108	Updating Tuner on endpoint	AUDIT

Log ID	Log message for Tuner Administrator	Severity level
35109	Failed to update Tuner on endpoint	MINOR
35110	Successfully updated Tuner on endpoint	AUDIT
35111	Restarting Tuner on endpoint	AUDIT
35112	Updating Inventory Service on multiple endpoints	AUDIT
35113	Updating Subscription Service on multiple endpoints	AUDIT
35114	Updating Tuner on multiple endpoints	AUDIT
35115	Restarting Tuner on multiple endpoints	AUDIT
35116	Applying General settings to Tuner	AUDIT
35117	Failed to apply General settings to Tuner	MAJOR
35118	Successfully applied General settings to Tuner	AUDIT
35119	Applying Security settings to Tuner	AUDIT
35120	Failed to apply Security settings to Tuner	MAJOR
35121	Successfully applied Security settings to Tuner	AUDIT
35122	Applying Custom Properties settings to Tuner	AUDIT
35123	Failed to apply Custom Properties settings to Tuner	MAJOR
35124	Successfully applied Custom Properties settings to Tuner	AUDIT
35125	Applying Advanced settings to Tuner	AUDIT
35126	Failed to apply Advanced settings to Tuner	MAJOR
35127	Successfully applied Advanced settings to Tuner	AUDIT
35128	Applying Schedule settings to channel	AUDIT
35129	Failed to apply Schedule settings to channel	MAJOR
35130	Successfully applied Schedule settings to channel	AUDIT
35131	Applying Security settings to channel	AUDIT
35132	Failed to apply Security settings to channel	MAJOR
35133	Successfully applied Security settings to channel	AUDIT
35134	Subscribing to channel	AUDIT
35135	Failed to subscribe to channel	MAJOR

Log ID	Log message for Tuner Administrator	Severity level
35136	Stopping channel	AUDIT
35137	Failed to stop channel	MAJOR
35138	Starting channel	AUDIT
35139	Failed to start channel	MAJOR
35140	Starting channel with arguments	AUDIT
35141	Failed to start channel with arguments	MAJOR
35142	Updating channel	AUDIT
35143	Failed to update channel	MAJOR
35144	Updating channel from a URL	AUDIT
35145	Failed to update channel from URL	MAJOR
35150	Unsubscribing from channel	AUDIT
35151	Failed to unsubscribe from channel	MAJOR
35152	Deleting channel	AUDIT
35153	Failed to delete channel	MAJOR
35154	Internal Tuner Admin error	AUDIT
35155	Cleaning up Tuner session	AUDIT
35156	Failed to initialize the Tuner session	MAJOR
35157	Failed to initialize the Tuner admin	MAJOR
35158	Failed to initialize the channel manager	MAJOR
35159	Failed to initialize the Tuner configuration	MAJOR
35160	Failed to initialize channel	MAJOR
35161	Using credentials to subscribe to channel	AUDIT
35162	Invalid credentials provided for subscribing to channel	MINOR
35163	Enabling SSL for administration port	AUDIT
35164	Enabling SSL for administration port failed	MAJOR
35165	Enabling SSL for administration port succeeded	AUDIT
35166	Disabling SSL for administration port	AUDIT
35167	Changing Tuner administration credentials	AUDIT

Log ID	Log message for Tuner Administrator	Severity level
35168	Changing Tuner administration port	AUDIT
35169	Changing Tuner administration port SSL settings	AUDIT
35170	Committing administration port changes	AUDIT
35171	Commit administration port changes failed	MAJOR
35172	Commit administration port changes succeeded	AUDIT
35173	Admin privileges required to perform a multiple-target action	MAJOR
35174	Tuner Admin session timed out	MAJOR
35175	Already subscribing or is subscribed to the channel	AUDIT
35176	Cancelling updating the channel	AUDIT
35177	Failed to cancel updating the channel	MAJOR
35178	Updating Transmitter on endpoint	AUDIT
35179	Successfully updated Transmitter on endpoint	AUDIT
35180	Failed to update Transmitter on endpoint	MAJOR
35181	Updating Proxy on endpoint	AUDIT
35182	Successfully updated Proxy on endpoint	AUDIT
35183	Failed to update Proxy on endpoint	MAJOR
35184	Updating Transmitter on multiple endpoints	AUDIT
35185	Updating Proxy on multiple endpoints	AUDIT
35189	Updating Patch on multiple endpoints	AUDIT
35190	Applying Common reboot settings to Tuner	INFO
35191	Failed to apply Common reboot settings to Tuner	MINOR
35192	Successfully applied Common reboot settings to Tuner	INFO
35196	Waking up the endpoint machines	INFO
35197	Started wake-up process on target machines	INFO
35198	Failed to wake-up endpoint machines	MAJOR

Tuner Packager

This section lists the log IDs, corresponding log messages, and log severity levels for Tuner Packager.

Log ID	Log message for Tuner Packager	Severity level
7100	Created data directory	AUDIT
7101	Source Tuner is located in directory	AUDIT
7102	Including alternate JRE	AUDIT
7103	Including alternate certdb file	AUDIT
7104	<console>	AUDIT
7105	Adding file to installer	AUDIT
7106	Preparing channel archive	AUDIT
7107	Including product license	AUDIT
7108	Packaging complete	AUDIT
7109	Channel directory preparation complete	AUDIT
7200	Copy error, source file does not exist	WARNING
7201	Error adding file to installer, file has 0 length	WARNING
7202	Widget mismatch	WARNING
7300	Error, an exception occurred	MINOR
7301	Error, channel source does not exist	MINOR
7302	Error, channel copy failed	MINOR
7303	Error, channel not found	MINOR
7304	Error, channel has no segments	MINOR
7305	Error, transmitter error	MINOR
7306	Error, transmitter busy	MINOR
7307	Error, transmitter not found	MINOR
7308	Error, license file does not exist	MINOR
7400	A license error occurred	MAJOR
7401	Packaging failed	MAJOR
7402	Publishing failed	MAJOR

Chapter

9 Ports

You can use ports to access Marimba components. For example, to access Report Center, you open a browser window and enter a URL that has this form:

`http://<machine_name>:<port>`

where `<machine_name>` is the name of the machine where Report Center is installed, and `<port>` is the port number for accessing Report Center.

The following topics are provided:

- List of ports (page 442)

List of ports

The following table lists the port names that Symphony Marimba Client Automation components use and the default settings for those ports.

Table 9-1: Ports

Port name	Component	Default settings
Database Server	Inventory Plug-in	1433 (SQL Server)
	Report Center	1521 (Oracle)
Deployment Manager Access	Browsers	8000
	Endpoint tuners and servers	
Directory Service	Transmitters	389 (non-SSL)
	Policy Manager	3268 (non-SSL)
	Deployment Manager	636 (SSL)
	Report Center	3269 (SSL)
Proxy Listening	Endpoint tuners	8080
Report Center/CMS Access	Report Center	8888
SNMP Port	Transmitter Administration	161
SNMP Manager Port	Transmitter Administration	162
SMTP Server	Deployment Manager	
	Transmitter Guardian	
Thread Dumper Listening	Browsers	5287
Transmitter Listening	Endpoint tuners	5282
	Publisher	
	Channel Copier	
	Repeaters and Mirrors	

Table 9-1: Ports (Continued)

Port name	Component	Default settings
Tuner RPC	Tuner Administrator	7717
	Transmitter Administrator	
	Deployment Manager	
	Publisher	
	Mirrors	
Lite Weight Tuner Administrator” Listening	Endpoint Tuners	7799

Index

Numerics

7717. See RPC port, tuner's.

A

- abortPublish command-line option 72
- abortUpdate command-line option 68
- access, restricting administration 104
- ACL base DN 299
- Action Request
 - log messages 312
 - active.name, property 248
 - ADAM schema, generating from command line 138
 - adapter.debug 270
 - adapter.state, property 262
 - adapter.update.minor.post.repair 271
 - adapter.update.post.repair 272
 - adapter.updateinstall 271
 - adapter.updateinstall.pending 271
 - adapter.updateinstall.silent 271
 - adapter.version, property 263
- add command-line option 122
- addCustomBulletin command-line option 71
- addCustomPatch command-line option 70
- addf command-line option 123
- addPatchGroupScript command-line option 72
- addPatchSource command-line option 68
- addPatchToPatchGroup command-line option 72
- addQueryToPatchGroup command-line option 72

- admin command-line option 27, 147
- admin options 146
- admin user name and password 64
- admin, property 248
- administration
 - permissions, setting 103, 104
 - port, proxy's. See RPC port, tuner's.
 - user name and password 103
- administration tools, log messages 313
- adminperms command-line option 104
- adminRDN command-line option 148
- allow.arbitrary parameter 243
- allowrunexe 272
- allProperties command-line option 147
- anonymize command-line option 27
- anonymous command-line option 146
- anyuser command-line option 146
- appExtUrl command-line option 126
- Application Packager
 - channel states 304
 - command-line options 36
 - log messages 314
- applytemplate command-line option 53
- appScanSchedule command-line option 126
- assignment states for a patch group 89
- author, property 248

B

- batch or -b command-line option 60
- bindaddr command-line option 143
- bind command-line option 148

blackout periods and the marimba.schedule.filter property 186

blackout schedules

- format for command line 99

BMC Software, contacting 2

browser.url, property 249

-bulletinSummary command-line option 72

C

-cabfile command-line option 49

cache

- changing the location of 112, 118
- low watermark 105
- maximum size 105
- repairing 111
- verifying 116

cache.lowWaterMark property 232

cache.maxSize property 232

-cachelowwm command-line option 105

-cacheReport command-line option 116

-cachesize command-line option 105

-cancelButton command-line option 126

-capabilities command-line option 123

capabilities, property 249

CAR files, installing 29

category, property 249

certificate

- configuring the console to use SSL 63, 66
- SSL 115, 121

Certificate Manager

- command-line options 56
- certPassword command-line option 28
- chain or -c command-line option 57
- chaining proxies 108, 109, 118
- changeorder command-line option 76

channel

property

- active.name 248
- adapter.state 262
- adapter.version 263
- admin 248
- author 248
- browser.url 249
- capabilities 249
- category 249

channel.version 249

channelmanager.rightclickmenuprefix 263

channelmanager.status.subscribed 263

channeltype 263

classpath 249

code 250

codebase 250

copyright 250

delete.alert 250

description 250

dsl.addtodsl 264

dsl.description 264

dsl.manufacturer 263

dsl.patchlastbuildid 264

dsl.product 264

dsl.version 264

edit.time 264

extension 250

filepackager.savemanifest 250

filepackager.setnewtimestamp 251

filepackager.setotherattributes 251

frame.bounds 251

height 251

hide 251

icon 252

icon.smaller 252

index.page 252

install.active 252

install.inactive 252

interrupt 252

iphost.expire 253

locale 253

logoff.action 253

logoff.args 253

logoff.notify 253

logon.action 253

logon.args 253

logon.notify 253

logs.arguments 254

logs.enabled 264

logs.roll.policy 265

logs.roll.size 265

logs.roll.versions 265

main 254

- mimetype 254
- name 254
- network.notify 254
- notifyremove 255
- pickup.tunerprops 255
- platform 255
- presentation 255
- publish.time 255
- receiveBPS 267
- restartTuner 267
- schedule.args 265
- security.sslOnly 256
- service.autostart.order 256
- service.daemon 256
- service.on-demand.name 256
- signing.certkey 256
- signing.enabled 256
- signing.scope 256
- start.schedule 257
- start.schedule.skipfirst 257
- subscription.schedule 257
- subscriptionmanager.push.patchserviceu
rl 266
- subscriptionmanager.push.subscriptions
erviceurl 266
- thumbnail 257
- thumbnail.running 257
- title 257, 265
- type 257
- undo.count 258
- update.action 258
- update.active 258
- update.check 258
- update.inactive 258
- update.schedule 258
- update.vary 259
- update.vary.max 259
- width 259
- proxy properties 228
- URL of Proxy Administrator 103
- windows.icon 259
- channel archive (CAR) files, installing 29
- Channel Copier 50
 - command-line options 59
- channel operation, command-line options 36
- channel parameters 270
- channel segments
 - locale codes 261
 - platform codes 260
- channel states
 - Application Packager 304
 - overview 303
- channel.version, property 249
- channelmanager.rightclickmenuprefix,
property 263
- channelmanager.status.subscribed, property 263
- channelname command-line option 48
- channels
 - locale codes 261
 - log ID ranges 311
 - operating system codes 260
 - platform codes 260
 - running from the command line 54
 - states 303
 - upgrading channels packaged with previous
Application Packager versions 52
- channeltype, property 263
- child containers
 - command for specifying 87
- classpath, property 249
- clear20PatchEdits command-line option 74
- clearcache command-line option 106
- clearPatchEdits command-line option 74
- clearsslpw command-line option 116, 141
- clientcertpw command-line option 77, 60, 123
- code, property 250
- codebase, property 250
- codes, logging 307
- Collections base DN 299
- command syntax 20
- command-line options
 - .NET Packager 43
 - AIX Patch Source options 75
 - Application Packager 36
 - Certificate Manager 56
 - Channel Copier 59
 - channel operation 36
 - Common Management Services (CMS) 62
 - console 62
 - Custom Application Packager 37, 40

- File Packager 38
 - overview 19
 - Patch Manager 66
 - Patch Service options 75
 - Patch Source options for AIX 75
 - PDA Packager 47
 - Policy Manager 76
 - Proxy Administrator 103
 - Proxy Server 116
 - Publisher 121
 - Report Center 125
 - running packaged channels 54
 - Schema Manager 137
 - specifying child containers 87
 - transmitter 141
 - Transmitter Administrator 145
 - Tuner Administrator 157
 - tuners 27
 - Windows Installer Packager 42
- Common Management Services (CMS)
- command-line options 62
 - log messages 328
 - concurrency command-line option 106
 - Config base DN 299
 - configSet command-line option 77
 - configure command-line option 117
 - console
 - command-line options 62
 - console command-line option 28
- Content Replicator
- log messages 333
 - copyPatchGroup command-line option 72
 - copyright, property 250
 - createPatchGroup command-line option 73
- Custom Application Packager
- command-line options 37, 40
- customer support 3
- custompackage command-line option 37
- D**
- D command-line option 78
 - database command-line option 147
 - database schema installation from command line 139
 - database schema reinstallation from command line 140
 - database schema uninstall from command line 140
 - database schema upgrade from command line 141
 - database server port 442
 - dbHost command-line option 126
 - dbName command-line option 126
 - dbPort command-line option 127
 - dbPw command-line option 127
 - dbtree_password 300
 - dbType command-line option 127
 - dbUser command-line option 127
 - debugOnConsole command-line option 28
 - delayfiledownload 272
 - delete command-line option 79, 60
 - delete or -d command-line option 57
 - delete.alert, property 250
 - deletePatchSourcePatches command-line option 68
 - deletePluginFromDb command-line option 128
 - deleting policies, using the command line 79
 - delPatchGroup command-line option 73
 - delPatchSource command-line option 68
 - delPatchToPatchGroup command-line option 73
 - dependChannelSilent 272
 - dependencyFile command-line option 68
 - depends.update 273
 - Deployment Manager access port 442
 - Deployment Manager, log messages 336
 - Deployment Service, log messages 336
 - description, property 250
 - detailedHelp command-line option 67
 - dialog.timeouts 273
 - directory for logs 107
 - directory service port 442
 - directory services
 - assigning directory service-to-repeater mappings 82
 - disableProcessRedirect 273
 - disableSSL command-line option 63
 - disk-based queue on transmitter 131
 - display command-line option 28
 - display options 147
 - dlltimeout 273
 - domain command-line option 148
 - dotnetpackage command-line option 44

-downloadAllPatches command-line option
 downloadAllPatches commandline option] 69
 dsl manufacturer property 263
 dsl.addtodsl property 264
 dsl.description property 264
 dsl.patchlastbuildid property 264
 dsl.product property 264
 dsl.version property 264
 -dst command-line option 60

E

edit.time, property 264
 editor.save.history.prompt.suppress 273
 -emergency command-line option 146
 -enableSSL command-line option 63
 encryption, using SSL for 63, 66
 endpoint
 listing policies assigned 91
 -exec command-line option 54
 exit codes 25
 -export command-line option 80, 69
 -export or -e command-line option 57
 -export_dbtree command-line option 137
 -export20PatchEdits command-line option 74
 exporting policies from the command line 80
 -exportPatchEdits command-line option 74
 -exportPatches command-line option 69
 -exportQueries command-line option 128
 extension, property 250

F

-f command-line option 142
 File Packager
 command-line options 38
 filebyref command-line argument 48
 -filepackage command-line option 38
 filepackager.savemanifest, property 250
 filepackager.setnewtimestamp, property 251
 filepackager.setotherattributes, property 251
 -fileStorageReport command-line option 143
 fileswithchannel parameter 274
 -filter command-line option 69
 -forceInstall command-line option 75
 -forcePublish command-line option 73
 -forcePublishPatchGroup command-line

option 73
 -forceRemove command-line option 75
 -forceStage command-line option 75
 forceterm parameter 274
 forcetimeout parameter 274
 frame.bounds, property 251
 -fsck command-line option 143

G

-generate_ldap_schema command-line
 option 138
 -getBindAddress command-line option 63
 -getconfig command-line option 107
 -getLogDir command-line option 63
 -getMaxConn command-line option 63
 -getPatchGroupScript command-line option 73
 -getPort command-line option 63
 -getProperty command-line option 147
 -getRoot command-line option 63
 -getSSLCert command-line option 64
 -getSystemProp command-line option 67
 grant/deny
 not showing 172
 -group command-line option 146

H

-h (help) command-line option 142, 64, 128
 height, property 251
 -help command-line option 23, 67, 123
 -help or -h command-line option 57, 61
 Hidden entries 299
 hide, property 251
 host name
 of the proxy 103, 109, 118
 -httpport command-line option 143
 HTTPS 112, 118

I

icon, property 252
 icon.smaller, property 252
 -ignore command-line option 123
 -import command-line option 80, 70
 -import or -i command-line option 57
 -import_dbtree command-line option 138
 -import20PatchEdits command-line option 75

- importCollections command-line option 129
 - importing policies from the command line 80
 - importPatchEdits command-line option 74
 - importQueries command-line option 130
 - index.page, property 252
 - indexStorageReport command-line option 143
 - infraExtUrl command-line option 130
 - infraScanSchedule command-line option 130
 - Infrastructure Service
 - log messages 362
 - inheritTunerPassword command-line option 146
 - inifileobjects 275
 - install command-line option 54, 75
 - install.active, property 252
 - install.inactive, property 252
 - install.priority 275
 - install_db_schema command-line option 139
 - installchannel command-line option 28
 - installing CAR files 29
 - installing database schema from command line 139
 - internal transmitters 112, 118
 - interrupt, property 252
 - intro command-line option 29
 - Inventory
 - log messages 370
 - iphost.expire property 253
- J**
- java command-line option 29
 - Java command-line options 29
 - jit command-line option 30
- K**
- keeprunning command-line option 30
- L**
- ldap options 148
 - LDAP Query Collections base DN 299
 - ldapservers command-line option 82
 - list command-line option 84
 - list or -l command-line option 57
 - listener port, proxy's 109, 118
 - listing policies from the command line 84
 - load balancing 112, 118
- local proxies, administering 103
 - locale codes 261
 - locale command-line option 30
 - locale, property 253
 - location
 - of cache, moving 112, 118
 - of logs, moving 107
 - log IDs
 - overview 307
 - severity level examples 309
 - log information
 - Action Request 312
 - administration tools 313
 - Application Packager 314
 - Common Management Services (CMS) 328
 - Content Replicator 333
 - Deployment Manager 336
 - Deployment Service 336
 - Infrastructure Service 362
 - Inventory 370
 - log ID ranges 311
 - Logging 376
 - Marimba Migration Module 378
 - Patch Management 380
 - Patch Manager 380
 - Patch Service 385
 - Policy Management 386
 - Proxy 397
 - Proxy Administrator 399
 - Report Center 400
 - Schema Management 407
 - setup and deployment 409
 - storage 414
 - Subnet Repeater Policy 415
 - Transmitter Administrator 424
 - transmitters 416
 - Tuner Administrator 434
 - Tuner Packager 369, 438
 - tuners 426
 - log.directory property 233
 - log.roll.policy property 234
 - log.roll.size property 234
 - log.roll.versions property 234
 - logdir command-line option 107
 - logging information 307

logging queue on the transmitter 131
 Logging, log messages 376
 -logo command-line option 30
 logoff.action, property 253
 logoff.args, property 253
 logoff.notify, property 253
 logon.action, property 253
 logon.args, property 253
 logon.notify, property 253
 logs
 changing the location of 107
 properties for changing in the proxy 234
 logs.arguments, property 254
 logs.enabled, property 264
 logs.roll.policy, property 265
 logs.roll.size, property 265
 logs.roll.versions, property 265
 low watermark 105

M

Machine import base DN 300
 -machines command-line option 85
 machines flat file, importing 85
 -main command-line option 147
 main options 148
 main, property 254
 main.clientCertPassword property 239
 main.transmitter.root property 239
 mapping directory services to repeaters 82
 mapping file 82
 Marimba Migration Module, log messages 378
 marimba.auditlog.enabled 167
 marimba.bandwidth.max 167
 marimba.browser.home 167
 marimba.browser.url 168
 marimba.channelmanager.startmenu 168
 marimba.console.codepage 168
 marimba.dialup.enabled 168
 marimba.dialup.password. 169
 marimba.dialup.server 169
 marimba.dialup.servers 169
 marimba.dialup.timeout 169
 marimba.dialup.user. 169
 marimba.http.agent 169
 marimba.http.port 170

marimba.http.showtip 170
 marimba.intro.url 170
 marimba.inventory.concat.tunerid 170
 marimba.inventory.customscanner.timeout 170
 marimba.inventory.nativescanner.timeout 171
 marimba.inventory.plugin.dbinsert 171
 marimba.inventory.plugin.enablepatchhistory 17
 1
 marimba.inventory.plugin.forwardURL 171
 marimba.inventory.scanner.prop.scanner.comp.so
 ftwaretitle.scan property 172
 marimba.inventory.smbiosbiosversion
 property 172
 marimba.launch.args 172
 marimba.launch.console 173
 marimba.launch.console.hide 173
 marimba.launch.defaultWakeuptime 173
 marimba.launch.inheritPath 173
 marimba.launch.javaArgs 174
 marimba.launch.logFile 174
 marimba.launch.logLevel 174
 marimba.launch.maxWakeuptime 174
 marimba.launch.multiplecpu 175
 marimba.launch.NTServiceArgs 175
 marimba.launch.redirect 175
 marimba.launcher.autorestart 175
 marimba.ldap.admanagementdomain 176
 marimba.ldap.connectiontimeout 176
 marimba.ldap.querytimeout 176
 marimba.ldap.srvdnsserver 177
 marimba.logs.enabled 177
 marimba.logs.rangefilter 177
 marimba.logs.roll.policy 178
 marimba.logs.roll.size 178
 marimba.logs.roll.versions 178
 marimba.network.detection.address 178
 marimba.network.detection.delay.offline 179
 marimba.network.detection.delay.online 179
 marimba.network.detection.ping.retries 179
 marimba.network.detection.policy 180
 marimba.primary.url 180
 marimba.proxy.bypass 180
 marimba.proxy.enable 181
 marimba.proxy.exceptions 181
 marimba.proxy.http.host 181

marimba.proxy.http.list 181
marimba.proxy.http.password 182
marimba.proxy.http.table 182
marimba.proxy.https.host 181
marimba.proxy.https.list 181
marimba.proxy.https.password 182
marimba.proxy.https.table 182
marimba.proxy.notforunqualifiedhosts 183
marimba.proxy.socks.host 183
marimba.proxy.trust 183
marimba.proxy.url.bypassalways 184
marimba.reboot.aix.soft 184
marimba.reboot.interact 184
marimba.reboot.interact.allowSnooze 184
marimba.reboot.interact.snooze.maxTime 184
marimba.reboot.interact.timer 185
marimba.reboot.never 185
marimba.schedule.dialup 185
marimba.schedule.filter 186
marimba.schedule.filter.property 186
marimba.schedule.polite 186
marimba.schedule.polite.speed 187
marimba.schedule.quietupdates 187
marimba.schedule.restrict 187
marimba.schedule.startdelay 187
marimba.security.cdrom 187
marimba.security.cert.password.timeout 187
marimba.security.channels.onlytrusted 188
marimba.security.clientcertpw 188
marimba.security.cookies 188
marimba.security.credentials.cache 188
marimba.security.identity.transmitters 188
marimba.security.logging 188
marimba.security.multicast.access 188
marimba.security.noUserOverride 189
marimba.security.printing 189
marimba.security.rpc.access 189
marimba.security.ssl.matchdomainonly 189
marimba.security.sslcert 190
marimba.security.trusted.certs 190
marimba.security.trusted.transmitters 190
marimba.subscribe 191
marimba.subscription.adminusers 193
marimba.subscription.dolast 193
marimba.subscription.installmode 193
marimba.subscription.inventory.compliance 194
marimba.subscription.machinename 194
marimba.subscription.nodelete 194
marimba.subscription.reapplyconfigonfail 195
marimba.subscription.reboot.allowcancel 195
marimba.subscription.retrycount 195
marimba.subscription.retryintervalsec 195
marimba.subscription.retrytime 195
marimba.subscription.timeout 196
marimba.subscription.update.schedule 196
marimba.subscription.updatealways 196
marimba.subscription.usecomputername 196
marimba.subscription.useshortcuts 196
marimba.subscription.usexml 197
marimba.subscription.varytime 197
marimba.tuner.admin 197
marimba.tuner.admin.emergency 198
marimba.tuner.admin.ldap.basedn 198
marimba.tuner.admin.ldap.binddn 199
marimba.tuner.admin.ldap.countlimit 199
marimba.tuner.admin.ldap.enable 199
marimba.tuner.admin.ldap.groupclass 199
marimba.tuner.admin.ldap.groupexcludecla 199
marimba.tuner.admin.ldap.groupmemberattr 199
marimba.tuner.admin.ldap.groupnameattr 200
marimba.tuner.admin.ldap.onlyreturnusers 200
marimba.tuner.admin.ldap.ouattr 200
marimba.tuner.admin.ldap.ouclass 200
marimba.tuner.admin.ldap.password 200
marimba.tuner.admin.ldap.searchdns 201
marimba.tuner.admin.ldap.server 202
marimba.tuner.admin.ldap.ssl 202
marimba.tuner.admin.ldap.type 202
marimba.tuner.admin.ldap.useouforgroups 203
marimba.tuner.admin.ldap.userclass 203
marimba.tuner.admin.ldap.useridattr 203
marimba.tuner.display.nocancel 203
marimba.tuner.display.noerrors 203
marimba.tuner.display.noprogress 204
marimba.tuner.display.nosavedpasswords 204
marimba.tuner.display.nowarnings 204
marimba.tuner.display.version 204
marimba.tuner.enabletaskbaricons 205
marimba.tuner.iphost.expire 205
marimba.tuner.iphost.refresh 206

marimba.tuner.keeprunning 207
 marimba.tuner.logs.applyFilters 208
 marimba.tuner.logs.centralizedlogging 208
 marimba.tuner.logs.platform.eventsource 208
 marimba.tuner.logs.platform.logname 209
 marimba.tuner.logs.platformlogging 209
 marimba.tuner.logs.roll.policy 209
 marimba.tuner.logs.roll.size 210
 marimba.tuner.logs.roll.versions 210
 marimba.tuner.minimize 210
 marimba.tuner.minimize.log 210
 marimba.tuner.name 210
 marimba.tuner.network.httpTimeout 211
 marimba.tuner.nt.reflect.username 212
 marimba.tuner.nt.username.persist 213
 marimba.tuner.outgoing.host 213
 marimba.tuner.p2p.bcast.addr 211
 marimba.tuner.p2p.enabled 211
 marimba.tuner.p2p.port 211
 marimba.tuner.receipt.maxdays 213
 marimba.tuner.receipt.maxsize 213
 marimba.tuner.release.version 214
 marimba.tuner.restart.timeout 214
 marimba.tuner.rpc.cert.id 214
 marimba.tuner.rpc.host 214
 marimba.tuner.rpc.pool 215
 marimba.tuner.rpc.port 215
 marimba.tuner.rpc.secure 215
 marimba.tuner.rpc.ssllisten 215
 marimba.tuner.rpc.sslport 215
 marimba.tuner.status.enable 215
 marimba.tuner.trayicon.menu.about.enabled 216
 marimba.tuner.trayicon.menu.cancel.enabled 216
 marimba.tuner.trayicon.menu.exit.enabled 216
 marimba.tuner.trayicon.menu.open.enabled 217
 marimba.tuner.trayicon.tooltip.normal.text 217
 marimba.tuner.trayicon.tooltip.receiving.text 217
 marimba.tuner.uninstall.allowed 217
 marimba.tuner.update.credentials 217
 marimba.tuner.update.name 217
 marimba.tuner.update.periodic 217
 marimba.tuner.update.periodic.silent 218
 marimba.tuner.update.profile 218
 marimba.tuner.update.restart 218
 marimba.tuner.update.schedule 218

marimba.tuner.update.seturl 219
 marimba.tuner.update.unsigned 219
 marimba.tuner.url 219
 marimba.tuner.user 219
 marimba.tuner.workspace.dir 219
 -maxDbConn command-line option 130
 MESH (multi-endpoint synchronized host) tuner properties 211
 mimetype, property 254
 -minDbConn command-line option 130
 -minimal command-line option 30
 moving
 the cache directory 112, 118
 the logs directory 107
 MRBA_XXX_growth 301
 MRBA_XXX_maxsize 301
 MRBA_XXX_size 301
 msi.commandline.<mode> 275
 msi.commandline.<mode>.launchflags 276
 msi.commandline.<mode>.use 276
 msi.db 276
 msi.download 276
 msi.download.delete 277
 msi.download.path 277
 msi.install 277
 msi.install.args 277
 msi.installed.reinstall 278
 msi.log.attr 278
 msi.log.mode 278
 msi.log.path 278
 msi.patch.change.reinstall 278
 msi.policy.elevated 278
 msi.policy.enabled 279
 msi.policy.ldbrowse 279
 msi.policy.log 279
 msi.policy.nobrowse 279
 msi.policy.nomedia 280
 msi.policy.norollback 280
 msi.product 280
 msi.product.advertise 280
 msi.product.installed 280
 msi.redirect 281
 msi.repair.mode 281
 msi.repair.source.download 282
 msi.source.alternate 282

msi.transform.change.reinstall 282
msi.transform.path 283
msi.ui 283
msi.ui.basic.progressonly 283
msi.ui.flags 283
msi.ui.raw 284
msi.upgrade.uninstall 284
msi.upgrade.uninstall.checkresult 285
-multicast command-line option 30

N

name, property 254
-namespace command-line option 87
natok parameter 244
.NET Packager, command-line options 43
network.notify, property 254
-nickname command-line option 57
nobackup 285
-nodisplay command-line option 28
nofilemap 285
-nointro command-line option 29
-nojit command-line option 30
-nologo command-line option 30
-nomulticast command-line option 30
-nopprimary command-line option 31
-noproxychain command-line option 108
normal proxy 109
-normalproxy command-line option 109
-norpc command-line option 31
-nosecurity command-line option 31
nostoragecompact 286
-nosubscribe command-line option 23
notifyremove 286
notifyremove, property 255
-noupdate command-line option 24, 33
noverifychecksums 286
noverifyspace 286
-nowin command-line option 30

O

offline patching
 command to obtain data from Internet 69
operating system codes 260
options, command-line 19
-output command-line option 123

P

-p (primary channel) command-line option 23
package properties
 setting from the command line 96
packaged applications, running from the
 command line 54
-packagedir command-line option 48
-packagePatch command-line option 71
packages
 states 304
parameters
 channel 270
 fileswithchannel 274
 forceterm 274
 forcetimeout 274
 rundir 292
 runuser 293
parameters, channel 270
parameters.txt file 270
-paramfile command-line option 61
-passwd command-line option 146, 148
-password command-line option 87
password
 administration 103, 104
 client certificate 77
 specifying for command-line publish
 operations 91
 SSL certificate 114, 119
-password command-line option 64, 123, 131
patch groups
 assignment states 89
Patch Management
 log messages 380
Patch Manager
 command-line options 66
 log messages 380
Patch Service
 log messages 385
Patch Service options 75
Patch Source channels
 command-line option for AIX 75
patch.<x> 287
patch.args.<string> 287
patch.count 287
-patchsubscribe command-line option 87

-patchSummary command-line option 71
 path, directory 107, 112, 119
 PDA Packager
 command-line options 47
 -pdapackage command-line option 47
 permissions, administration 104
 pickup.tunerprops, property 255
 platform codes 260
 platform, property 255
 plug-in, Subscription
 publishing 90
 -pluginLogRollPolicy command-line option 131
 -pluginLogRollSize command-line option 131
 -pluginLogRollVersions command-line
 option 131
 -pluginMaxDiskSize command-line option 131
 -pluginMaxFiles command-line option 132
 -pluginState command-line option 132
 policies
 deleting from the command line 79
 listing from the command line 84
 policy
 listing endpoint assignments 91
 Policy Management
 deleting policies from the command line 79
 exporting policies from the command line 80
 importing policies from the command line 80
 log messages 386
 Policy Manager, command-line options 76
 port
 proxy's listener 109, 118
 tuner's RPC 103
 -port command-line option 109, 117
 ports 441
 postscripts.fail 287
 preload 287
 presentation, property 255
 -preview command-line option 76, 123
 -primary command-line option 31
 primary schedule
 format for command line 102
 -print or -p command-line option 57
 processCreationFlags 288
 product support 3
 -progressBar command-line option 132
 -prompt command-line option 132
 properties
 proxy 228
 transmitters 238
 tuner launch arguments 172
 properties, channel 248
 properties.txt file 248
 -propsfile command-line option 61, 124
 proxies
 administration port. See RPC port, tuner's.
 cache of 105
 chaining 108, 109, 118
 host name 103, 109, 118
 normal 109
 port
 administration. See RPC port, tuner's.
 listener 109, 118
 properties 228
 reverse 112, 118
 secure reverse 113
 starting to accept requests 115
 stopping from accepting requests 115
 Proxy
 log messages 397
 Proxy Administrator
 channel URL of 103
 command-line options 103
 log messages 399
 proxy listening port 442
 Proxy Server
 command-line options 116
 proxy.adminPermissions property 229
 proxy.enableTaskBarIcons property 229
 proxy.inheritTunerPassword property 230
 proxy.outgoingHost property 230
 proxy.root property 230
 -proxychain command-line option 118
 -publish command-line option 90, 147
 publish options 150
 publish.time, property 255
 Publisher, command-line options 121
 publishing the Subscription plug-in 90
 -publishPatchGroup command-line option 73
 -publishpw command-line option 91
 -pubPw command-line option 133

-pubUser command-line option 133

Q

-q (quiet) command-line option 124
queryCheckUpdate 289
-quiet or -q command-line option 57
-quit command-line option 23, 31

R

-range command-line option 133
reboot.allow 289
reboot.allowcancel 290
reboot.force 290
reboot.showdialog 290
reboot.wait.seconds 290
receiveBPS, property 267
-redirect command-line option 31
refresh.credentials property 111, 233
-refreshcache command-line option 110
-refreshsourcelist command-line option 54
-reinstall_db_schema command-line option 140
reinstalling database schema from command line 140
remote proxies, administering 103
-remove command-line option 55, 36
-repackage command-line option 39, 43, 46, 48
-repair command-line option 55
repair schedule

format for the command line 102

-repaircache command-line option 111
repairing the cache 111
repeater.schedule.restricted property 239, 240
repeater.sync.blackout property 240
-repeaterInsert command-line option 133
repeating options 151

Report Center
command-line options 125

log messages 400

Report Center/CMS access port 442

reporter
listing policies assigned to endpoints 91
-reporter command-line option 91
-reset command-line option 67
-resetDatabase command-line option 70
restartTuner, property 267

restricting access to the proxy 104

return codes 25

reverse proxy, target of 112, 118

-reverseproxy command-line option 112, 118

-revertPatchGroup command-line option 74

rollback 291

-roottdir command-line option 112, 118, 144

-rpc command-line option 23, 31

RPC port, tuner's 103

run 291

run.install 293

run.update 293

runargs 291

runchannel

exit codes 25

program 22

return codes 25

-runCollection command-line option 133

rundetach 291

-rundir command-line option 55

rundir parameter 292

-runexe command-line option 55

runtime.ssl.provider 221

runtime.tuner.intelamt 223

runtime.verbose 223

runuser parameter 293

S

savedelay 293

-savesslpw command-line option 119, 142

sbrp -export 154

sbrp -import 154

-scan command-line option 134

scanner.services.rcstartmsgCMD.enable property 256

-scannerState command-line option 134

-schedule command-line option 134

schedule.args, property 265

schedules

blackout format for command line 99

format for command line, primary or secondary schedule 102

format for specifying 99

format for the command line, repair schedule 102

- format for the command line, update schedule 102
- schema base DN 300
- schema command-line option 148
- Schema Management
 - log messages 407
- Schema Manager
 - command-line options 137
- scripts.perm parameter 294
- secondary schedule
 - format for command line 102
- secure command-line option 113
- Secure Sockets Layer (SSL)
 - configuring the console to use 63, 66
- security command-line option 31
- security.sslOnly, property 256
- segment command-line option 124
- segment locale codes 261
- segment platform codes 260
- semisilent 294
- semisilent.cancel.disable 294
- server command-line option 148
- server.connect.active property 230
- server.connect.backlog property 230
- server.connect.ipaddr property 230
- server.connect.port property 230
- server.connect.rate property 231
- server.proxychain property 231
- server.proxychain.nextProxy property 231
- server.reverse property 231
- server.reverse.secure property 231
- server.reverse.secure.certid property 231
- server.reverse.secure.certpw property 231
- server.reverse.secure.clientAuth property 231
- server.reverse.secure.savepw property 232
- server.reverse.secure.strongEncryption property 232
- server.reverse.target property 232
- server.transfer.buflen property 232
- service, running the tuner as a service 34
- service.autostart.order, property 256
- service.daemon, property 256
- service.ondemand.name, property 256
- serviceType command-line option 134
- setAcl command-line option 135
- setAdminPassword command-line option 64
- setBindAddress command-line option 64
- setDownloadPolicy command-line option 67
- setLogDir command-line option 65
- setmacro command-line option 55
- setMaxConn command-line option 65
- setNoSSL command-line option 65
- setpluginparam command-line option 92
- setPort command-line option 65
- setProperty command-line option 113, 111
- setreferencetime 295
- setRoot command-line option 66
- setSSL command-line option 66
- setsslpw command-line option 114, 120, 142
- setSystemProp command-line option 67
- setup and deployment
 - log messages 409
- setUpdateSchedule command-line option 68
- setuse or -s command-line option 58
- severity command-line option 135
- show command-line option 32
- showBulletin command-line option 72
- showDownloadPolicy command-line option 68
- showfail 295
- showFilters command-line option 70
- showMachinesByPatch command-line option 70
- showPatch command-line option 71
- showPatchGroup command-line option 74
- showPatchGroups command-line option 74
- showPatchProps command-line option 74
- showPatchSources command-line option 70
- showProductVersions command-line option 70
- showSystemProps command-line option 68
- shrinkwrappackage command-line option 41
- sign command-line option 61, 124
- signfile command-line option 124
- signing.certkey, property 256
- signing.enabled, property 256
- signing.scope, property 256
- silent command-line option 55
- silent 295
- simulate 295
- simulate command-line option 74
- SMTP server port 442
- snapshot command-line option 41

SNMP Manager Port 442
SNMP Port 442
snmp.manager.port 224
-softwareUsage command-line option 135
-sourcedir command-line option 48
spaces, in the directory path 107, 112, 119
-src command-line option 61
SSL certificate, setting passwords for 115, 121
-sslcert command-line option 114
-ssljava command-line option 32
-sslpw command-line option 115, 121, 144
-stagemsi command-line option 56
-start command-line option 115, 32
start.schedule, property 257
start.schedule.skipfirst, property 257
starting
 the proxy to accept requests 115
 the tuner, command-line options 19
states
 channels 303
 for patch groups 89
static IP address
 problems with 180
stats.days property 240
stats.enabled property 240
stats.report.enabled property 241
stats.report.http.url property 241
stats.report.interval property 241
stats.report.type property 241
-status command-line option 115, 135
-stop command-line option 115, 36
stopping the proxy from accepting requests 115
storage, log messages 414
-strategy command-line option 153
subnet repeater policy 243, 244
Subnet Repeater Policy, log messages 415
subnet-based repeater policy (SBRP) 154
-subPw command-line option 135
-subscribe command-line option 94, 23, 36
subscribing targets to packages from the command line 94
subscribing targets to patch groups from the command line 87
Subscription base DN 300
Subscription config base DN 300

Subscription plug-in
 publishing 90
subscription.schedule, property 257
subscriptionmanager.push.patchserviceurl, property 266
subscriptionmanager.push.subscriptionserviceurl, property 266
-subUser command-line option 136
support, customer 3
syntax, command 20
-sysExtUrl command-line option 136
-sysScanSchedule command-line option 136

T

target
 proxy (for chaining) 109, 118
 transmitters 112, 118
technical support 3
-testCerts command-line option 32
thread dumper listening port 442
thumbnail, property 257
thumbnail.running, property 257
-timeout command-line option 24
title, property 257, 265
-tmp command-line option 62
-trans command-line option 147
trans options 154
transform.<x> 296
transform.count 295
transmitter
 command-line options 141
Transmitter Administrator
 command-line options 145
 log messages 424
transmitter listening port 442
transmitter.http.bindAddress property 242
transmitter.http.externalURL property 242
transmitter.http.outgoing.host property 242
transmitter.logs.useclientip property 242
transmitters
 disk-based queue on 131
 host name 112, 118
 internal 112, 118
 log messages 416
 properties 238

- target 112, 118
- triggerOn command-line option 136

- trust command-line option 32

Tuner Administrator

- command-line options 157

- log messages 434

- tuner command-line option 96, 24

Tuner Packager

- log messages 369, 438

- tuner program 27

tuner properties

- setting from the command line 96

- tuner RPC port 443

tuners

- command-line options 27

- log messages 426

- RPC port 103

- running as services 34

- starting arguments 172

- txadminaccess command-line option 98

- type, property 257

U

- undo.count, property 258

- uninstall_db_schema command-line option 140

- uninstalling database schema from command line 140

- uninstallolockedfile 296

- unloadname command-line argument 49

- unpublish command-line option 124

- unsign command-line option 62

- unsubscribe command-line option 36

- update command-line option 24, 33

- update command-line option 36

update schedule

- format for the command line 102

- update.action, property 258

- update.active property (deprecated) 258

- update.check property 258

- update.inactive, property 258

- update.schedule, property 258

- update.vary property 259

- update.vary.max property 259

- updateDatabase command-line option 70

- updateInstall command-line option 76

- upgrade command-line option 52

- upgrade_db_schema command-line option 141

- upgradePatchEdits command-line option 75

- upgradePatchRepository command-line option 75

- upgrading channels packaged with previous Application Packager versions 52

- upgrading database schema from command line 141

- url command-line option 136

- URL of Proxy Administrator 103

- user command-line option 99, 66, 124, 137, 147

- user name, administration 103, 104

- userenv 296

- userobjs 297

- users options 155

V

- v (verbose) command-line option 33, 124

- verbose or -v command-line option 62

- verify command-line option 56

- verify.schedule 298

- verifycache command-line option 116

- verifying the cache 116

- verifyrepair.update.install 298

- version command-line option 53, 56

- view or -v command-line option 58

W

- w command-line option 99

- width, property 259

- Windows Installer Packager, command-line options 42

- Windows service, running the tuner as a service 34

- windows.icon, property 259

- wipackage command-line option 42

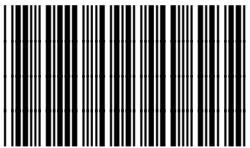
- WMI (Windows Management Instrumentation)

- using -scan command-line option for 134

- workspace options 156

- ws (workspace directory) command-line option 24, 33

- wts 298



439164