Foglight® 6.0.0

Command-Line Reference Guide

their respective owners.

Legend

■ | **WARNING: A WARNING icon indicates a potential for property damage, personal injury, or death.**

! | **CAUTION: A CAUTION icon indicates potential damage to hardware or loss of data if instructions are not followed.**

i | **IMPORTANT NOTE**, **NOTE**, **TIP**, **MOBILE**, or **VIDEO:** An information icon indicates supporting information.

Foglight Command-Line Reference Guide
Updated - May 2021
Software Version - 6.0.0

Contents

1

About the Command-Line Interface

This *Command-Line Reference Guide* provides information about the Foglight® command-line interface. You can use Foglight commands to interface with different components of your monitoring environment instead of the browser interface.

This guide is intended for Foglight System Administrators who are new to Foglight can find information about syntax conventions along with an overview of all Foglight commands in the first chapter of this guide. The following chapters contain reference information about specific commands, such as the Foglight Management Server and Foglight Agent Manager commands. Advanced users can find information about the `fglcmd` interface in the last chapter. The last chapter contains information about the commands that provide administrative-level commands. Read this chapter to find out how to deploy and create agents, install and activate cartridges, manage licenses, and do many other administrative tasks from the command line.

This manual describes the commands that come with your Foglight installation and allow you to access different components in your monitoring environment without having to use the standard graphical interface such as the Administration dashboards. This can be useful in situations when you need to either automate your business logic by writing scripts, or use regular expressions to select multiple entities where applicable.

- Looking at Foglight Commands
- Getting Started with Foglight Commands

Syntax Conventions

This manual employs a set of conventions that are used in the command syntax. You should review them before getting started with the manual in order to successfully make use of Foglight commands in your monitoring environment. The syntax conventions are as follows:

- Generic examples follow the UNIX path structure that uses forward slashes '/' to separate directories.

- Platform-specific examples follow standard platform conventions. For example, UNIX-specific examples use forward slashes '/' as directory delimiters, while Windows examples use backslashes '\'.

- *<foglight_home>* is a placeholder that represents the path to the Foglight Management Server installation.

- *<foglight_agent_mgr_home>* is a placeholder that represents the path to the Foglight Agent Manager installation. This can be the location of the Foglight Agent Manager installation on a monitored host, or the home directory of the Foglight Agent Manager that comes embedded with the Foglight Management Server. For example:

   Path to the Foglight Agent Manager installation on a monitored host (Windows):

   C:\Quest\Foglight_Agent_Manager

   Path to the embedded Foglight Agent Manager installation (Windows):

   C:\Quest\Foglight\fglam

- Unless otherwise specified, Foglight commands are case-sensitive.

Looking at Foglight Commands

Foglight provides a number of commands that you can use to manage different components in your monitoring environment. The commands reside in the following directories of your Foglight Agent Manager or Foglight Management Server installation.

Foglight Management Server

- *<foglight_home>/bin*: Contains commands that manage the Foglight Management Server and administer agents, cartridges, metrics, and other entities. For more information about these commands, see Managing the Foglight Management Server on page 12 and Managing Agents, Cartridges, and Metrics on page 35.

- *<foglight_home>/tools:* Contains the Remote Monitor utility as well as the administrative commands in ZIP files. For more information about the Remote Monitor, see the Foglight *Installation and Setup Guide*; for information on the Remote Monitor command syntax, see remotemonitor on page 19.

  If required, you can extract these commands to a remote computer in your network and issue them from there. For more information, see Managing the Foglight Management Server on page 12 and Getting Started with Administrative Commands on page 42.

Foglight Agent Manager

- *<foglight_agent_mgr_home>/bin*: Contains commands that manage the Foglight Agent Manager. For more information, see Managing the Foglight Agent Manager on page 26.

For more information about the *<foglight_home>* and *<foglight_agent_mgr_home>* placeholders, see Syntax Conventions on page 7.

The following table lists Foglight commands.

Table 1. Foglight commands

Component	Domain	Command	Description
Foglight Management Server	Server	fms	Provides command-line interface to the Foglight Management Server process
		fmsha	Provides command-line interface to the Foglight Management Server process running in High Availability (HA) mode
		remotemonitor	Provides command-line interface to the Remote Monitor application
	Database	foglight_db_upgrade	Verifies the database version and upgrades it to a higher version if necessary
		runDB	Starts the embedded database
		shutdownDB	Stops the embedded database
	Authentication	keyman	Provides an interface that lets you change, add, or delete encryption keys, or generate application tokens

Table 1. Foglight commands

Component	Domain	Command	Description
Foglight Agent Manager	Agent Manager	fglam	Provides command-line interface to the Foglight Agent Manager
		support-bundle	Generates Agent Manager support bundles
Administrative Interface	Agents	activate	Activates one or more Foglight agent instances
		clients	Shows a list of Foglight Agent Manager instances
		clientupgrade	Upgrades one or more remote instances of the Foglight Agent Manager
		create	Creates one or more Foglight agent instances
		deactivate	Deactivates one or more Foglight agents
		delete	Deletes one or more Foglight agent instances
		deploy	Deploys one or more Foglight agent packages
		getlog	Retrieves a copy of an agent log file
		list	Shows a list of Foglight agent instances
		logs	Shows a list of agent log files
		packages	Shows a list of available agent packages
		setschedule	Assigns a blackout schedule to one or more Foglight agent instances
		showschedule	Shows the blackout schedule assigned to one or more Foglight agent instances.
		start	Starts the data collection for one or more Foglight agent instances
		stop	Stops the data collection for one or more Foglight agent instances
		types	Shows a list of Foglight agent types sorted by agent manager ID

Table 1. Foglight commands

Component	Domain	Command	Description
Administrative Interface (continued)	Cartridges	disable	Deactivates a Foglight cartridge
		enable	Activates an installed Foglight cartridge
		install	Installs a cartridge on the Foglight Management Server
		list	Lists all installed Foglight cartridges
		uninstall	Uninstalls a Foglight cartridge
	Security	assigngroup	Adds or removes a user from a group
		assignrole	Adds or removes a role from a group
		creategroup	Creates a group
		createuser	Creates a user
		deletegroup	Deletes a group
		deleteuser	Deletes a user
		exportldapcfg	Exports the LDAP settings to an XML file
		importldapcfg	Imports the LDAP settings from an XML file
		list	Lists users, groups, and roles
	Licenses	import	Installs a Foglight license
		list	Shows a list of installed Foglight licenses
		remove	Removes a Foglight license using the license serial number
Administrative Interface (continued)	Utility	blackoutobject	Assigns a blackout schedule to topology objects
		blackouts	Lists the blackout schedules assigned to topology objects
		bundle	Generates a support bundle file
		configexport	Exports the monitoring policy to an XML file
		configimport	Imports the monitoring policy
		env	Outputs the values of server configuration parameters
		metricexport	Exports metric observations to a file using a metric query
		list	Shows a list of all Foglight schedules
		run	Runs a script
		topologyexport	Exports the value of one or more properties of a topology object to an XML file
		uiexport	Exports a UI module into a ZIP file
		uiimport	Imports a UI module from a ZIP file
		uilist	Shows a list of deployed UI modules

Getting Started with Foglight Commands

To get started with Foglight commands:

ℹ | **NOTE:** These instructions describe a simple configuration in which the Foglight™ Management Server and the Foglight Agent Manager are installed on the same platform.

1   Ensure that you have all of the following components installed in your monitoring environment.

A typical setup includes the following components:

- Foglight Management Server
- Foglight Agent Manager
- Infrastructure Cartridge

For complete information on installing the above components, see the *Installation and Setup Guide*.

2   Obtain your Foglight user name and password from the Foglight Administrator.

3   Ensure that your `JAVA_HOME` system variable points to the location of JRE that comes with Foglight. For more information about the JRE version, see the appropriate *Installation and Setup Guide*.

4   Start the Foglight Management Server and the Foglight Agent Manager.

For example, on Windows® platforms, you can start the Foglight Management Server by choosing **Start > Programs > Quest > Foglight 6.0.0 > Start Foglight**.

To start the Foglight Agent Manager, issue the `fglam` command. The `fglam` command can found in the *<foglight_agent_mgr_home>/bin* directory. For complete information about the `fglam` command, see fglam on page 27.

5   If you experience problems starting either the Foglight Management Server or the Foglight Agent Manager, complete the following steps:

a   Check the log files. Foglight stores the log file in the following locations:

Foglight Management Server

<foglight_home>/**logs/ManagementServer_**_date_time_file_ID_.**log**

Foglight Agent Manager

<foglight_agent_mgr_home>/**state/default/logs/FglAM_**_date_time_file_ID_.**log**

Where

- *date* and *time* are the date and time the Foglight Management Server or Foglight Agent Manager were started
- *file_ID* identifies the log file

For information about *foglight_home* and *foglight_agent_mgr_home*, see Syntax Conventions on page 7.

b   If you do not find any indicators as to why the agent is not collecting data, refer to the *Release Notes* and *Administration and Configuration Help* for more information.

c   If you do not find any relevant information in the documentation, contact Quest Support. For more information, see About Us on page 112.

From here, you can proceed to any of the following sections, as required:

- Managing the Foglight Management Server on page 12
- Managing the Foglight Agent Manager on page 26
- Managing Agents, Cartridges, and Metrics on page 35

Managing the Foglight Management Server

Foglight offers a set of commands that allow you to perform server-related operations through the command-line interface. You can use these commands to perform a variety of tasks, such as start or stop the Foglight Management Server, run or upgrade the database, or manage encryption keys.

The majority of server-specific commands access the server directly. As such, you can only issue them on the computer that is running an instance of the Foglight Management Server. One exception is the Remote Monitor utility that you can use to communicates with multiple instances of the Foglight Management Server running in High Availability (HA) mode. For more information about running the Foglight Management Server running in HA mode, see fmsha on page 16.

Your Foglight installation includes a compressed file, *<foglight_home>/tools/remotemonitor.zip*, that contains the Remote Monitor utility. If required, you can extract the contents of that file to a remote computer in your Foglight network and run the Remote Monitor utility from that computer, as described in To get started with Foglight Management Server commands: on page 12. For information about the location of *foglight_home*, see Syntax Conventions on page 7.

- Getting Started
- Running the Server in Stand-Alone Mode
- Running the Server in High Availability Mode
- Starting the Embedded Database
- Stopping the Embedded Database
- Upgrading the Database
- Managing Encryption Keys

Getting Started

To get started, make sure the Foglight Management Server is running, and, if you are planning to access the server commands remotely, copy and extract the contents of the *remotemonitor.zip* file to your remote computer.

For more information about the Remote Monitor, see remotemonitor on page 19.

To get started with Foglight Management Server commands:

ℹ️ | **NOTE:** This procedure continues from Getting Started with Foglight Commands on page 11.

1   Copy the *<foglight_home>/tools/remotemonitor.zip* file from the computer that has a running instance of the Foglight Management Server to the remote computer.

2   Extract the contents of the *remotemonitor.zip* file to a directory on the remote computer. For example, if you want to run the Remote Monitor on a Foglight Agent Manager computer, extract the *remotemonitor.zip* file to the *<foglight_agent_mgr_home>/bin* directory.

3   Reference the directory that contains the Foglight Management Server commands, either *<foglight_home>/bin* or *<foglight_agent_mgr_home>/bin*, by completing one of the following steps:

- If you want to use the command prompt, open a Command Prompt window (Windows®) or a terminal window (UNIX® or Linux®) and navigate to *<foglight_home>/bin* or *<foglight_agent_mgr_home>/bin* as required.

OR

- If you want to use a Foglight Management Server command in a script, ensure that your script references *<foglight_home>/bin* or *<foglight_agent_mgr_home>/bin* as applicable.

For more information about *foglight_home* and *foglight_agent_mgr_home*, see Syntax Conventions on page 7.

From here, you can proceed to any of the following procedures as required:

Running the Server in Stand-Alone Mode

fms

The `fms` command provides command-line interface to the Foglight Management Server process. It offers a set of options that allow you to perform any of the following operations as required:

- Start or stop the Foglight Management Server in stand-alone mode

- Install and start the Foglight Management Server in stand-alone mode as a Windows® service

- Stop and remove a Foglight Management Server Windows service

- Configure Java™ Virtual Machine (JVM) options and add entries to the Foglight classpath

- Assign different names to different Foglight Management Server process launchers

- Display version information or a list of arguments along with their descriptions

Syntax

Windows only

```
fms [-s|--start|-Dquest.debug=debug_level] [-q|--stop] [-w|--wait]
   [-n|--name process_name] [-i|--install-service] [-r|--remove-service]
   [-b|--start-service] [-j|--jvm-argument JVM_options]
   [-p|--classpath class_path] [-v|--version] [-h|--help] [-t|--thread-dump]
   [JVM_option] [-m|--javavm path_to_JAVA_HOME]
   [--set-global-debug-level debug_level]
   [--add-debug-level package_1.=debug_level_1 … package_n.=debug_level_n]
   [--remove-debug-level package_1. … package_n.]
```

Unix only

```
fms [-d|--daemon] [-s|--start|-Dquest.debug=debug_level] [-q|--stop] [-w|--wait]
    [-n|--name process_name] [-j|--jvm-argument JVM_options]
    [-p|--classpath class_path] [-v|--version] [-h|--help] [-t|--thread-dump]
    [JVM_option] [-m|--javavm path_to_JAVA_HOME]
    [--set-global-debug-level debug_level]
    [--add-debug-level package_1.=debug_level_1 … package_n.=debug_level_n]
    [--remove-debug-level package_1. … package_n.]
```

> **i** | **NOTE:** If you do not specify any options, fms uses the default option, s, and starts an instance of the
> Foglight Management Server.

Table 1. Options and arguments

Options		Argument	Description
Unix and Windows			
add-debug-level		package.=debug_l evel	Specifies a debug level for one or more packages. Higher debug levels result in more detailed logging.
			For example, you can set the *debug_level* to one of the following values:
			• **0**: No debugging
			• **1**: Minimal debugging
			• **2**: Detailed debugging
			The period '.' following the package name is mandatory, otherwise the package is treated as a class. The *package.=debug_level* argument can be specified multiple times.
h	**help**	None	Displays a list of arguments and their descriptions.
-Dquest.debug=*debug_level*			Runs the Foglight Management Server in debug mode. For example, you can set the *debug_level* to one of the following values:
			• **0**: No debugging
			• **1**: Minimal debugging
			• **2**: Detailed debugging
			By default, this option outputs the logs to the standard output. You can redirect the output to a file if required. For example:
			`fms -Dquest.debug=3 > debug3.log`
j	**jvm-argument**	*JVM_options*	Specifies one or more JVM options.
JVM_option			Passes an option to the JVM, prefixed with **-X** or **-D**. For example:
			`fms -Xdebug`
			`-Xrunjdwp:transport=dt_socket,server=y,`
			`suspend=n,address=1044 -Dquest.debug=2`
m	**javavm**	*path_to_JAVA_HO ME*	Points to the JVM for the Management Server process.
n	**name**	*process_name*	Specifies a unique process name for the current instance of the Foglight Management Server. Foglight uses process names to distinguish between different instances of the same process launcher.
p	**classpath**	*class_path*	Adds entries to the JVM classpath.
q	**stop**	None	Stops the running Foglight Management Server process.

Table 1. Options and arguments

Options		Argument	Description
remove-debug-level		*package.*	Removes a debug level associated with one or more packages. The period '.' following the package name is mandatory, otherwise the package is treated as a class. The *package.* argument can be specified multiple times.
s	start	None	Starts the Foglight Management Server.
set-global-debug-level		*debug_level*	Sets the global debug level. The *debug_level* argument must be a non-negative integer. Higher debug levels result in more detailed logging. For example, you can set the *debug_level* to one of the following values: • **0**: No debugging • **1**: Minimal debugging • **2**: Detailed debugging
t	thread-dump	None	Requests a thread output from the running application. This option writes the output to a separate log file in the application's installation directory.
v	version	None	Displays the version number, copyright, build number, and the installation directory.
w	wait	None	When sending a shutdown command to an existing Foglight Management Server process, this option instructs the command to wait indefinitely for the process to exit before shutting it down.
Unix only			
d	daemon	None	Starts the Foglight Management Server as a daemon process.
Windows only			
b	start-service	None	Starts the Foglight Management Server Windows service.
i	install-service	None	Installs the Foglight Management Server as a Windows service.
r	remove-service	None	Stops and removes the Foglight Management Server Windows service.

Examples

Displaying version information

```
C:\Quest\Foglight\bin>fms -v
Foglight 6.0.0
   Copyright (c) 2017 Quest Software Inc.
   Build Number: 6.0.0-201703280645-e47f1f70-1464
   Client Binary Directory: C:\Quest\Foglight
```

Starting the server in stand-alone mode

```
C:\Quest\Foglight\bin>fms -s
5/13/21-10-16 11:18:46.807 INFO Starting Foglight Management Server.
5/13/21-10-16 11:18:46.901 INFO Foglight Management Server starting up using
   database host: localhost
5/13/21-10-16 11:18:46.916 INFO Foglight Management Server starting up using
   database port: 13306

5/13/21-10-16 11:18:46.916 INFO Foglight Management Server starting up using
   database name: foglight

...
```

Stopping the server running in stand-alone mode

```
C:\Quest\Foglight\bin>fms -q
```

i | **NOTE:** This command displays a series of messages in the Command Prompt window (Windows) or the terminal window (Unix or Linux) instance used to stop the Foglight Management Server. These messages indicate that the server is stopping.

See also

-
-

Running the Server in High Availability Mode

fmsha

The `fmsha` command provides a command-line interface to the Foglight Management Server process running in High Availability (HA) mode. Running Foglight in HA mode allows you to manage multiple instances of the Foglight Management Server in a JBoss® partition that supports the HA feature. For more information about the HA mode, see the *Installation and Setup Guide*.

This command offers a set of options that you can use to perform any of the following operations:

- Start or stop the Foglight Management Server in HA mode
- Install and start the Foglight Management Server in HA mode as a Windows® service
- Stop and remove a Foglight Management Server HA Windows service
- Configure Java™ Virtual Machine (JVM) options and add entries to the Foglight classpath
- Assign different names to different Foglight Management Server process launchers
- Display version information or a list of arguments along with their descriptions

Syntax

Windows only

```
fmsha [-s|--start] [-q|--stop] [-w|--wait] [-n|--name process_name]
   [-i|--install-service] [-r|--remove-service] [-b|--start-service]
   [-j|--jvm-argument JVM_options] [-p|--classpath class_path] [-v|--version]
   [-h|--help] [-t|--thread-dump] [-m|--javavm path_to_JAVA_HOME]
   [JVM_option] [--set-global-debug-level debug_level]
   [--add-debug-level package_1.=debug_level_1 … package_n.=debug_level_n]
   [--remove-debug-level package_1. … package_n.]
```

Unix only

```
fmsha [-d|--daemon] [-s|--start] [-q|--stop] [-w|--wait]
   [-n|--name process_name] [-j|--jvm-argument JVM_options]
   [-p|--classpath class_path] [-v|--version] [-h|--help] [-t|--thread-dump]
```

```
[-m|--javavm path_to_JAVA_HOME] [JVM_option]
[--set-global-debug-level debug_level]
[--add-debug-level package_1.=debug_level_1 … package_n.=debug_level_n]
[--remove-debug-level package_1. … package_n.]
```

> **i** | **NOTE:** If you do not specify any options, fmsha uses the default option, s, and starts an instance of the
> Foglight™ Management Server in HA mode.

Table 2. Options and arguments

Options		Argument	Description
Unix and Windows			
add-debug-level		package**.**=debug_l evel	Specifies a debug level for one or more packages. Higher debug levels result in more detailed logging. For example, you can set the *debug_level* to one of the following values: • **0**: No debugging • **1**: Minimal debugging • **2**: Detailed debugging The period '.' following the package name is mandatory, otherwise the package is treated as a class. The *package***.**=*debug_level* argument can be specified multiple times.
h	**help**	None	Displays a list of arguments and their descriptions.
j	**jvm-argument**	*JVM_options*	Specifies one or more JVM options.
JVM_option			Passes an option to the JVM, prefixed with **-X** or **-D**. For example: fmsha **-X**debug **-X**runjdwp:transport=dt_socket,server=y, suspend=n,address=1044 **-D**quest.debug=2
m	**javavm**	*path_to_JAVA_HO ME*	Points to the JVM for the Management Server process.
n	**name**	*process_name*	Specifies a unique process name for the current instance of the Foglight Management Server. Foglight uses process names to distinguish between different instances of the same process launcher.
p	**classpath**	*class_path*	Adds entries to the JVM classpath.
q	**stop**	None	Stops the running Foglight Management Server process.
remove-debug-level		*package***.**	Removes a debug level associated with one or more packages. The period '.' following the package name is mandatory, otherwise the package is treated as a class. The *package***.** argument can be specified multiple times.
s	**start**	None	Starts the Foglight Management Server.
set-global-debug-level		*debug_level*	Sets the global debug level. The *debug_level* argument must be a non-negative integer. Higher debug levels result in more detailed logging. For example, you can set the *debug_level* to one of the following values: • **0**: No debugging • **1**: Minimal debugging • **2**: Detailed debugging
t	**thread-dump**	None	Requests a thread output from the running application. This option writes the output to a separate log file in the application's installation directory.

Table 2. Options and arguments

Options		Argument	Description
v	version	None	Displays the version number, copyright, build number, and the installation directory.
w	wait	None	When sending a shutdown command to an existing Foglight Management Server process, this option instructs the command to wait indefinitely for the process to exit before shutting it down.
Unix only			
d	daemon	None	Starts the Foglight Management Server as a daemon process.

> **NOTE:** Unix installations also include the following scripts for stsarting and stopping the Management Server in HA mode:
> - *fmsStartupHA.sh*
> - *fmsShutdownHA.sh*

Windows only			
b	start-service	None	Starts the Foglight Management Server Windows service.
i	install-service	None	Installs the Foglight Management Server as a Windows service.
r	remove-service	None	Stops and removes the Foglight Management Server Windows service.

Examples

Starting the server in HA mode

```
C:\Quest\Foglight\bin>fmsha
5/13/21-10-16 12:06:32.632 INFO Starting Foglight Management Server with the command
   bin\fms -Dfoglight.cluster.mode=true...
5/13/21-10-16 12:06:33.398 INFO Starting Foglight Management Server.
5/13/21-10-16 12:06:33.538 INFO Foglight Management Server starting up using
   database host: localhost
...
```

Installing the server in HA mode as a Windows service

```
C:\Quest\Foglight\bin>fmsha -i
5/13/21-10-16 12:08:43.000 INFO [native] Foglight High Availability Management
   Server (FoglightHA) service installed
```

Removing the server HA Windows service

```
C:\Quest\Foglight\bin>fmsha -r
5/13/21-10-16 12:11:24.000 INFO [native] Removed the Foglight High Availability
   Management Server (FoglightHA) service installed from
   'C:\Quest\Foglight'
```

See also

remotemonitor

The `remotemonitor` command provides command-line interface to the Remote Monitor application. The Remote Monitor communicates with multiple instances of the Foglight Management Server running in HA mode. If a server fails to reply, the Remote Monitor logs an event and sends e-mails to the server administrator. The Remote Monitor uses a list of host names or host names and port numbers to identify High Availability servers that it communicates with. This information is stored in the *<foglight_home>/config/remote_monitor.config* file under the `server.urls` entry.

For information on how to install and configure the Remote Monitor, see Managing the Foglight Management Server on page 12; for additional information about the Remote Monitor application, see the *Installation and Setup Guide*.

The `remotemonitor` command offers a set of options that you can use to perform any of the following operations:

- Start or stop the Remote Monitor

- Install and start the Remote Monitor as a Windows® service

- Stop and remove the Remote Monitor Windows service

- Configure Java™ Virtual Machine (JVM) options and add entries to the Remote Monitor classpath

- Assign different names to different Remote Monitor process launchers

- Display version information or a list of arguments along with their descriptions

Syntax

Windows only

```
remotemonitor [-s|--start] [-q|--stop] [-w|--wait] [-n|--name process_name]
   [-i|--install-service] [-r|--remove-service] [-b|--start-service]
   [-j|--jvm-argument JVM_options] [-p|--classpath class_path] [-v|--version]
   [-h|--help] [-t|--thread-dump] [JVM_option]
```

Unix only

```
remotemonitor [-s|--start] [-q|--stop] [-w|--wait] [-n|--name process_name]
   [-j|--jvm-argument JVM_options] [-p|--classpath class_path] [-v|--version]
   [-h|--help] [-t|--thread-dump] [JVM_option]
```

> **i** | **NOTE:** If you do not specify any options, remotemonitor uses the default option, `s`, and starts an instance of the Remote Monitor utility.

Table 3. Options and arguments

Options		Argument	Description
Unix and Windows			
h	**help**	None	Displays a list of arguments and their descriptions.
j	**jvm-argument**	*JVM_options*	Specifies one or more Java Virtual Machine (JVM) options.
JVM_option			Passes an option to the JVM, prefixed with **-x** or **-D**. For example:
			`remotemonitor` **-X**`debug`
			-X`runjdwp:transport=dt_socket,server=y,`
			`suspend=n,address=1044` **-D**`quest.debug=2`

Table 3. Options and arguments

Options		Argument	Description
n	**name**	*process_name*	Specifies a unique process name for the current instance of the Foglight™ Management Server. Foglight uses process names to distinguish between different instances of the same process launcher.
p	**classpath**	*class_path*	Adds entries to the JVM classpath.
q	**stop**	None	Stops the running Foglight Management Server process.
s	**start**	None	Starts the Foglight Management Server.
t	**thread-dump**	None	Requests a thread output from the running application. This option writes the output to a separate log file in the application's installation directory.
v	**version**	None	Displays the version number, copyright, build number, and the installation directory.
w	**wait**	None	When sending a shutdown command to an existing Foglight Management Server process, this option instructs the command to wait indefinitely for the process to exit before shutting it down.
Windows only			
b	**start-service**	None	Starts the Foglight Management Server Windows service.
i	**install-service**	None	Installs the Foglight Management Server as a Windows service.
r	**remove-service**	None	Stops and removes the Foglight Management Server Windows service.

Examples

Installing Remote Monitor as a Windows service

```
C:\Quest\Foglight\bin>remotemonitor -i
5/13/21-10-16 12:26:46.000 INFO [native] RemoteMonitor (RemoteMonitor) service
installed
```

Removing the Remote Monitor Windows service

```
C:\Quest\Foglight\bin>remotemonitor -r
5/13/21-10-16 12:28:37.000 INFO [native] Removed the RemoteMonitor (RemoteMonitor)
service installed from 'C:\Quest\Foglight'
```

Displaying Remote Monitor version information

```
C:\Quest\Foglight\bin>remotemonitor -v
Foglight RemoteMonitor 6.0.0
  Copyright (c) 2017 Quest Software Inc.
  Build Number: 5.7.5.8-201703280645-e47f1f70-1464
  Client Binary Directory: C:\Quest\Foglight
```

See also

Starting the Embedded Database

runDB

The `runDB` command starts the embedded database.

Syntax

runDB

Options and arguments

None

Example

```
C:\Quest\Foglight\bin>runDB
5/13/21-10-16 12:31:27.924 INFO Starting up the embedded database...
5/13/21-10-16 12:31:31.266 INFO Embedded database successfully started
```

See also

Stopping the Embedded Database

shutdownDB

The `shutdownDB` command stops the embedded database.

Syntax

shutdownDB

Options and arguments

None

Example

```
C:\Quest\Foglight\bin>shutdownDB
5/13/21-10-16 12:32:42.539 INFO Checking if the embedded database still running...
5/13/21-10-16 12:32:46.599 INFO Successfully shutdown the embedded database
```

See also

-
-

Upgrading the Database

foglight_db_upgrade

The `foglight_db_upgrade` command verifies the database version and upgrades it to a higher version if necessary. This command is useful when you need to migrate data from a Foglight environment that is running a lower version of the embedded database. This typically happens when the database upgrade that is initiated through the installer fails to complete.

Syntax

foglight_db_upgrade [-config path_to_server.config] [**-debug**]

Table 4. Options and arguments

Options	Argument	Description
config	*path_to_server.config*	Provides the location to the *server.config* file.
debug	None	Prints debugging information.

Example

```
C:\Quest\Foglight\bin>foglight_db_upgrade -debug
2012-09-05 10:03:09.875 INFO Starting up the embedded database...
2012-09-05 10:03:13.281 INFO Checking if the embedded database still running...
2012-09-05 10:03:14.234 INFO Successfully shutdown the embedded database
2012-09-05 10:03:15.250 INFO STDIN/mysqld:120905 10:03:14 [Note] Plugin
  'FEDERATED' is disabled.
2012-09-05 10:03:20.250 INFO STDIN/mysqld:120905 10:03:19 InnoDB: Started; log
  sequence number 0 717068346
2012-09-05 10:03:20.250 INFO STDIN/mysqld:120905 10:03:19 [Note] Event
  Scheduler: Loaded 0 events
2012-09-05 10:03:20.250 INFO STDIN/mysqld:120905 10:03:19 [Note]
  C:\Quest\Foglight\mysql\bin\mysqld.exe: ready for connections.
2012-09-05 10:03:20.250 INFO STDIN/mysqld:Version: '5.1.45-enterprise-
  commercial-pro' socket: '' port: 13306 MySQL Enterprise Server - Pro
  Edition (Commercial)
2012-09-05 10:03:22.437 INFO Embedded database successfully started
2012-08-29 13:07:50.843 INFO STDIN/mysqld:120829 13:07:49 [Note] Plugin
  'FEDERATED' is disabled.
2012-08-29 13:07:51.843 INFO STDIN/mysqld:120829 13:07:50 InnoDB: Started; log
  sequence number 0 221724334
2012-08-29 13:07:51.843 INFO STDIN/mysqld:120829 13:07:50 [Note] Event
  Scheduler: Loaded 0 events
2012-08-29 13:07:51.843 INFO STDIN/mysqld:120829 13:07:50 [Note]
  C:\Quest\Foglight\mysql\bin\mysqld.exe: ready for connections.
2012-08-29 13:07:51.843 INFO STDIN/mysqld:Version: '5.1.45-enterprise-
  commercial-pro' socket: '' port: 13306 MySQL Enterprise Server - Pro
```

```
   Edition (Commercial)
2012-08-29 13:07:53.109 INFO Embedded database successfully started
...
```

See also

Managing Encryption Keys

Foglight stores encrypted passwords and uses application tokens to look up encryption keys. In a default installation, keystore and database passwords are encrypted using the `foglight.defaultkey`, stored in the *<foglight_home>/config/mstkey.properties* file.

In some cases, you may need to change the database password or the default keystore password, for extra security. A newly changed password must be encrypted using the keyman command, and updated in the appropriate configuration file:

- *Foglight database password* is stored encrypted in the *<foglight_home>/config/**server.config*** file, in one of the following configuration parameters:

   - `server.database.password` sets the password used by the Management Server to establish the connection with the database.

   - `server.database.embedded.password` sets the password of the embedded database. The Management Server uses this value to start the embedded database. This is usually the root password that can be changed and encrypted.

- *The <foglight_home>/config/**tomcat.keystore** password* is stored encrypted in the *<foglight_home>/server/tomcat/**server.xml*** file. The default keystore password is `nitrogen`.

To change, encrypt, and update a database password:

1 Open a Command Prompt window (Windows) or a terminal window (UNIX® or Linux®).

2 Encrypt the new password.

   a On the Management Server machine, navigate to the *<foglight_home>/bin* directory.

   b Issue the following command:

   keyman encpwd <*password*> foglight.defaultkey

   Where *password* is the newly changed database password.

   c Record the output of the `keyman` command. For example:

   Encrypted Password: q40799f927b44ba22192f3a4fa2f1cc91

   For more information about the `keyman` command, see keyman on page 24.

3 Update the *server.config* file with the newly encrypted database password.

   a On the Management Server machine, open the *<foglight_home>/config/server.config* file for editing.

   b In the *server.config* file, replace the value of the appropriate entry (`server.database.password` for external database or `server.database.embedded.password` for embedded database) with the encrypted password, as recorded in Step c.

   > **ℹ IMPORTANT:** The password must be enclosed in double quotes.

   c Save the changes to the *server.config* file and restart the Management Server.

d Change the password for the Management Server user in the database (the default user name is `foglight`). For more information, see your database documentation.

To change, encrypt, and update a tomcat.keystore password:

1 Open a Command Prompt window (Windows) or a terminal window (UNIX® or Linux®).

2 On the Management Server machine, navigate to the *<foglight_home>* directory and issue the following command:

```
jre/bin/keytool -J-server -storepasswd -keystore
    <keytool_path>/tomcat.keystore
```

    Where *keytool_path* is the directory path to the *tomcat.keystore* file.

    e Change the password for all of the keys within the keystore:

```
jre/bin/keytool -J-server -keypasswd -alias <key_name> -keystore
    <keytool_path>/tomcat.keystore
```

    Where:

    key_name is the name of the key in the keystore whose password you want to change.

    keytool_path is the directory path to the *tomcat.keystore* file.

> **i** | **TIP:** To list a keystore for all of its aliases, issue the following command:
> ```
> jre/bin/keytool -J-server -keystore <keytool_path>/tomcat.keystore -
> storepass <password> --list -v
> ```
> Where:
> *keytool_path* is the directory path to the *tomcat.keystore* file.
> `password` is the keystore password.

For more information about the `keytool` command, see your Oracle® Java™ documentation

3 Encrypt the new password.

    a On the Management Server machine, navigate to the *<foglight_home>/bin* directory.

    b Issue the following command:

```
keyman encpwd <password> foglight.defaultkey
```

    Where *password* is the newly changed *tomcat.keystore* password.

    c Record the output of the `keyman` command. For example:

    Encrypted Password: q40799f927b44ba22192f3a4fa2f1cc91

    For more information about the `keyman` command, see keyman on page 24.

4 Update the *server.xml* file with the newly encrypted *tomcat.keystore* password.

    a On the Management Server machine, open the *<foglight_home>/server/tomcat/server.xml* file for editing.

    b In the *server.xml* file, locate the `keystorePass` attribute, and update its value with the newly encrypted password, as recorded in Step c.

> **i** | **IMPORTANT:** The password must be enclosed in double quotes.

    c Save the changes to the *server.xml* file and restart the Management Sever.

keyman

The `keyman` command allows you to encrypt passwords, to change, add, or delete encryption keys, or to generate application tokens.

Syntax

```
keyman [chmstkey old_key new_key] [addappkey key_name key_text key_pwd]
   [delappkey key_name] [encpwd "pwd_str" key_name] [getapptkn key_name key_pwd]
```

Table 5. Options

Option	Description
addappkey	Adds an application key.
chmstkey	Changes the master key
delappkey	Deletes an application key.
encpwd	Encrypts a password string using an encryption key.
getapptkn	Generates an application token.

Table 6. Arguments

Argument	Description
key_name	Specifies the name of the application key.
key_pwd	Specifies the password that protects the key.
key_text	Specifies the key value.
new_key	Specifies the new key value.
old_key	Specifies the old key value.
pwd_str	Specifies the password value that is to be encrypted. **NOTE:** This value must always be enclosed in double quotes. For example: `keyman encpwd `**`"my_new_password"`**` foglight.defaultkey`

Example

```
C:\Quest\Foglight\bin>keyman addappkey my_key my_key_text my_key_pwd
KeyToken: f3f3d72d81b1959bdf32416357b57e97aa63b199
```

Managing the Foglight Agent Manager

The Foglight Agent Manager is an application that manages Foglight agents on monitored hosts and allows them to communicate with the Foglight Management Server.

Foglight offers a set of commands that allow you to perform Foglight Agent Manager operations through the command-line interface. You can use these commands to perform a variety of tasks, such as start or stop the Foglight Agent Manager, display the version information, manage JVM options, or create a Foglight Agent Manager support bundle.

The Foglight Agent Manager component manages agent instances and their communication with the Foglight Management Server. Foglight Management Server installs include an embedded Foglight Agent Manager. The embedded Foglight Agent Manager on the Foglight Management Server starts up and stops with the server. This agent manager instance can be used to deploy agents and monitor the host on which the Foglight Management Server is installed, if required. To monitor additional hosts in your environment, you must install an agent manager component separately on each host computer. For more information about installing agent managers on monitored hosts, see the *Agent Manager Guide.*

> **i** | **IMPORTANT:** Embedded Foglight Agent Manager instances do not automatically start on server startup if you are upgrading the server. You can override this behavior by setting the parameter `server.fglam.embedded` in *<foglight_home>/config/server.config* to `true`.

- Getting Started
- Starting or Stopping the Foglight Agent Manager
- Creating Agent Manager Support Bundles

Getting Started

To get started with Foglight Agent Manager commands:

> **i** | **NOTE:** This procedure continues from

- Navigate to the directory *<foglight_agent_mgr_home>/bin*.

  Where *foglight_agent_mgr_home* refers to the installation directory of the Foglight Agent Manager, either its installation directory on a monitored host, or the home directory of the Foglight Agent Manager that comes embedded with the Foglight Management Server.

  To do that, complete one of the following steps.

  - If you want to use the command prompt, open a Command Prompt window (Windows®) or terminal window (UNIX® or Linux®), and navigate to the appropriate directory.

  or

  - If you want to use a Foglight Agent Manager command in a script, ensure that your script references the appropriate directory.

Starting or Stopping the Foglight Agent Manager

fglam

The `fglam` command provides the command-line interface to the Foglight Agent Manager process. It offers a set of options that you can use to perform any of the following operations, as required:

- Start or stop the Foglight Agent Manager

- Install and start the Foglight Agent Manager as a Windows® service

- Stop and remove the Foglight Agent Manager Windows service

- Configure Java™ Virtual Machine (JVM) options and add entries to the Foglight Agent Manager classpath

- Assign a name to the Foglight Agent Manager process launcher

- Display version information or a list of arguments along with their descriptions

Syntax

Windows only

```
fglam [-s|--start|-Dquest.debug-debug_level] [-q|--stop] [-w|--wait]
   [-i|--install-service] [-r|--remove-service] [-b|--start-service]
   [-j|--jvm-argument JVM_options] [-m|--javavm java_path] [JVM_option]
   [-p|--classpath class_path] [-v|--version] [-h|--help] [-t|--thread-dump]
   [-u|--support-bundle] [-l|--location [state_directory]] [--check-connection]
   [--create-state] [[-C|--configure] [--headless|--silent] [--spid SPID_path]
   [--fms url={http|https}://host:port [proxy={http|https}://host:port
   proxy-user=USERNAME proxy-pass=PASSWORD proxy-ntlm-domain=DOMAIN]
   [--deletefms url={http|https}://host:port] [--deleteallfms] [--detectha]]
   [ssl-allow-self-signed={true|false}] [ssl-cert-common-name=CERTCOMMONNAME]
   [compressed] [address=ip_address] [--noservice]
   [--host-display-name host_name] [--reset-host-id]]
   [--set-global-debug-level level] [--add-debug-level package_name.=level]
   [--remove-debug-level package.] [--heap-dump file_name|--heap-dump-all
   file_name] [--add-certificate alias=path_to_certificate_file]
   [--delete-certificate alias] [--list-certificates] [[--downstream
   "<port=<port>,key-password=<password>>[,<host=<host>,type=<https|http>,
   host=<host>,size=Small|Medium|Large|Huge|Maximum>]"]|[--deletedownstream <port>]|
   [--deletealldownstream]] [--no-start-on-exit]
```

Unix only

```
fglam [-s|--start|-Dquest.debug-debug_level] [-q|--stop] [-w|--wait]
   [-j|--jvm-argument JVM_options] [-m|--javavm java_path] [JVM_option]
   [-p|--classpath class_path] [-v|--version] [-h|--help] [-t|--thread-dump]
   [-u|--support-bundle] [-l|--location [state_directory]] [--check-connection]
   [--create-state] [[-C|--configure] [--headless|--silent] [--spid SPID_path]
```

```
[--fms url={http|https}://host:port [proxy={http|https}://host:port proxy-
user=USERNAME proxy-pass=PASSWORD proxy-ntlm-domain=DOMAIN]
[--deletefms url={http|https}://host:port] [--deleteallfms] [--detectha]]
[ssl-allow-self-signed={true|false}] [ssl-cert-common-name=CERTCOMMONNAME]
[compressed] [address=ip_address] [--host-display-name host_name]
[--reset-host-id]] [--set-global-debug-level level]
[--add-debug-level package_name.=level] [--remove-debug-level package.]
[--heap-dump file_name|--heap-dump-all file_name]
[--add-certificate alias=path_to_certificate_file]
[--delete-certificate alias] [--list-certificates][[--downstream
"<port=<port>,key-password=<password>>[,<host=<host>,type=<https|http>,
host=<host>,size=Small|Medium|Large|Huge|Maximum>]"]|[--deletedownstream <port>]|
[--deletealldownstream]] [--no-start-on-exit]
```

Table 1. Options and arguments

Option		Argument	Description
Unix and Windows			
add-certificate		*alias=path_to_cert ificate_file*	Adds the certificate file to the collection of trusted certificates for this Agent Manager. The certificate is trusted for all SSL connections.
add-debug-level		*package_name.=l evel*	Enables debugging for the specified package. When debugging is enabled, debug messages are written to the log. The amount of debugging depends on the specified debugging level. The level must be a positive integer number. Higher debug levels result in more detailed logging. The argument can be specified multiple times with this option.
			NOTE: The period '.' between the package name and the debug level is mandatory. Failing to include it results in the package being treated as a class.
check-connection		None	Indicates if this instance of the Foglight™ Agent Manager currently has an upstream connection with the Foglight Management Server.
			NOTE: It can be used with the **location** option to query the connection status of other instances.
			Along with a command output that shows the details of the connection state, this option also returns an exit code to the OS, indicating the state of the upstream connection.
			The exit code can be viewed by issuing the following command:
			Windows:
			`echo %errorlevel%`
			Unix:
			`echo $?`
			An exit code of zero '0' indicates that the Foglight Agent Manager is successfully connected to the upstream server while a positive integer value indicates a connection disruption.
C	**configure**	None	Launches the Foglight Agent Manager installation and configuration interface, enabling the update of the existing configuration files.
			NOTE: The installation directory of the Foglight Agent Manager embedded on the Foglight Management Server does not include a JRE directory. When using this option to configure the embedded Foglight Agent Manager, you must also specify the path to the JVM using the **javavm** option. Failing to do so results in an error.
			If neither the **headless** nor the **silent** options are used, the **Agent Manager Installation and Configuration** dialog box appears, allowing you to configure the Foglight Agent Manager. For complete information, see the *Foglight Agent Manager Installation Guide*.

Table 1. Options and arguments

Option	Argument	Description
create-state	None	Creates a state directory and populates it with default configuration files. The process fails if the directory already exists.
		NOTE: Each Foglight Agent Manager instance can multiple state directories. On Unix systems, the entire Foglight Agent Manager installation, including all state directories, must be owned by the same system user.
		NOTE: Creating and using multiple state directories on the embedded Foglight Agent Manager, running on the Foglight Management Server host, is not supported. If you want to use multiple state directories on the embedded Foglight Agent Manager, disable the embedded Foglight Agent Manager (as documented in "Automatically Running the Embedded Foglight Agent Manager" in the *Foglight Agent Manager Installation Guide* set) and install a separate, externally controlled Foglight Agent Manager instance.
deleteallfms	None	When used with --**configure**, this option removes all Foglight Management Server URLs from *config.xml*.
delete-certificate	*alias*	Deletes the certificate file from the collection of trusted certificates for this Agent Manager, given the certificate alias specified when it was created. The certificate is no longer trusted for any SSL connections.
deletefms	**url=HTTP|HTTPS**:*//host:port*	When used with --**configure**, this option removes a Foglight Management Server URL from *config.xml*. The URL is specified as an argument, where *host* and *port* specify the host name of the machine on which the Foglight Management Server is installed, and the port number the server uses to communicate with the Foglight Agent Manager.
downstream	port=*port*	When used with --**configure**, this option creates a downstream connection.
		The **port** argument specifies the number of the port the Agent Manager uses to listen for downstream connections.
	key-password=*password*	This argument specifies the password needed to access the private key contained in the keystore.
	host=host	Specifies the host name to be set in the certificate.
	type=**http|https**	Specifies the type of the supported protocol.
	size**=Small|Medium|Large|Huge|Maximum**	Specifies the amounts of the disk and memory resources. **Small** allocates 10 MB for the disk queue and 512 MB of memory.
deletedownstream	port=*port*	When used with --**configure**, this option deletes a downstream connection given its port number.
deletealldownstream		When used with --**configure**, this option deletes all downstream connections.
detectha	None	When used with --**configure** and --**fms** pointing to a Management Server that is part of a High Availability (HA) cluster, this option instructs the installer to detect and test any available Management Server peers. Any peers that fail the test should be removed from *config.xml* to avoid populating this file with incorrect values.

Table 1. Options and arguments

Option	Argument	Description
-Dquest.debug-*debug_level*		Runs the Foglight Agent Manager in debug mode. You can set `debug_level` to one of the following values: • **1**: No debugging • **2**: Minimal debugging • **3**: Detailed debugging By default, this option prints the logs to the standard output. You can redirect the output to a file if required. For example: `fglam -Dquest.debug=2 > debug2.log`
fms		When used with --**configure**, the **fms** option can be used to specify the URL to the Foglight Management Server, using the arguments listed below.
	address=*ip_address*	The IP address of the Foglight Agent Manager that is used to connect with the Foglight Management Server.
	ssl-cert-common-name=*CERTCOMMONNAME*	The common name of the expected certificate.
	compressed	Enables HTTP compression.
	proxy=HTTP|HTTPS:*//host:port*	The URL to the proxy server (if applicable), where *host* and *port* specify the host name of the machine on which the proxy server is installed, and the port number the proxy server uses to communicate with the Foglight Agent Manager.
	proxy-ntlm-domain=*DOMAIN*	If proxy server is used for connection with the Foglight Management Server, *domain* specifies the Windows domain of the proxy server.
	proxy-pass=*PASSWORD*	If proxy server is used for connection with the Foglight Management Server, *password* specifies the user password for accessing the proxy server.
	proxy-user=*USERNAME*	If proxy server is used for connection with the Foglight Management Server, *user_name* specifies the user name for accessing the proxy server.
	ssl-allow-self-signed=true|false	Indicates if self-signed certificates are accepted (**true**) or not (**false**).
	url=HTTP|HTTPS:*//host:port*	The URL to the Foglight Management Server, where *host* and *port* specify the host name of the machine on which the Foglight Management Server is installed, and the port number the server uses to communicate with the Foglight Agent Manager.
h help	None	Displays a list of arguments and their descriptions.
headless	None	When used with --**configure**, it launches the Foglight Agent Manager installation and configuration interface on the command line. For complete information, see the *Foglight Agent Manager Installation Guide*.
heap-dump	*file_name*	Outputs a heap dump containing only live objects to a file.
heap-dump-all	*file_name*	Outputs a heap dump containing all objects to a file.
host-display-name	*host_name*	When used with --**configure**, it sets the Foglight Agent Manager display name. By default, the Foglight Agent Manager uses the host name that is automatically detected for the machine on which it is installed. In some cases you may need to use a different name, for example, when you need to use a different name that suits your business needs. For more information, see the *Agent Manager Guide*.
j jvm-argument	*JVM_options*	Specifies one or more Java Virtual Machine (JVM) options.

Table 1. Options and arguments

Option		Argument	Description

JVM_option			Passes an option to the JVM, prefixed with **-x** or **-D**. For example:
			`fglam` **-X**`debug`
			-X`runjdwp:transport=dt_socket,server=y,`
			`suspend=n,address=1044` **-D**`quest.debug=2`

list-certificates		None	Displays a list of certificates that are added to this Agent Manager. Certificates that are included in the JRE do not appear on the list.
l	**location**	*state_directory*	Sets the location of the state directory for this instance of the Foglight Agent Manager. A state directory contains the Foglight Agent Manager configuration files. If not provided, the command uses the default location of the state directory, *<foglight_agent_mgr_home>/state*. **NOTE:** Each Foglight Agent Manager instance can multiple state directories. On Unix systems, the entire Foglight Agent Manager installation, including all state directories, must be owned by the same system user. **NOTE:** Creating and using multiple state directories on the embedded Foglight Agent Manager, running on the Foglight Management Server host, is not supported. If you want to use multiple state directories on the embedded Foglight Agent Manager, disable the embedded Foglight Agent Manager (as documented in "Automatically Running the Embedded Foglight Agent Manager" in the *Foglight Agent Manager Installation Guide* set) and install a separate, externally controlled Foglight Agent Manager instance.
m	**javavm**	*java_path*	Sets the path to the JVM. **NOTE:** The platforms that use a Sun JRE require the JRE version 1.6.0_14 or higher. When this option points to an earlier version of the JRE, the Foglight Agent Manager fails to launch.
no-start-on-exit		None	If you want to install the Agent Manager Windows service, or a Unix deamon, the configuration interface appears with pre-selected options, indicating that the Agent Manager will start immediately after the installation. You can force these options to appear disabled by default (and enable them, if required, during the configuration), if you start the `fglam` command with this option (together with --**configure** and -- **headless**): `fglam --configure --headless` **--no-start-on-exit**
p	**classpath**	*class_path*	Adds entries to the JVM classpath.
q	**stop**	None	Stops the running Foglight Agent Manager process.
remove-debug-level		*package_name*.	Disables debugging for the specified package. When debugging is disabled, package-specific debug messages are no longer written to the log. The argument can be specified multiple times with this option. **NOTE:** The period '.' immediately after the package name is mandatory. Failing to include it results in the package being treated as a class.
reset-host-id		None	When used with --**configure**, it resets the unique ID that this Foglight Agent Manager instance uses to identify itself to the Foglight Management Server. **NOTE:** Do not reset the unique identifier unless Quest Support requests that you do so.

Table 1. Options and arguments

Option		Argument	Description
set-global-debug-level		*level*	Sets the global debugging level. When debugging is enabled, debug messages are written to the log. The amount of debugging depends on the specified debugging level. The level must be a positive integer number. Higher debug levels result in more detailed logging.
spid		*SPID_path*	During a silent installation, it migrates a Foglight Client to a new Foglight Agent Manager instance, given the path to the Foglight Client installation directory (*SPID_path*).
s	start	None	Starts the Foglight Agent Manager.
silent		None	When used with --**configure**, it launches the Foglight Agent Manager configuration, without user interaction, applying the default configuration values. For complete information, see the *Foglight Agent Manager Installation Guide*.
t	thread-dump	None	Requests a thread output from the running application. This option writes the output to a separate log file in the application's installation directory.
u	support-bundle	None	Requests a support bundle from the running application.
v	version	None	Displays the version number, copyright, build number, and the installation directory.
w	wait	None	When sending a shutdown command to an existing Foglight Agent Manager process, this option instructs the command to wait indefinitely for the process to exit before shutting it down.
Windows only			
b	start-service	None	Starts the Foglight Agent Manager Windows service.
i	install-service	None	Installs the Foglight Agent Manager as a Windows service.
noservice		None	When used with --**configure**, it prevents the Foglight Agent Manager from being installed as a Windows service.
r	remove-service	None	Stops and removes the Foglight Agent Manager Windows service.

Examples

Displaying version information

```
C:\Quest\Foglight_Agent_Manager\bin>fglam -v
Foglight Agent Manager 5.8.5.8
   Copyright (c) 2017 Quest Software Inc.
   Build Number: 5.8.5.8-201408261438-BETA-427
   Client Binary Directory: C:\Quest\Foglight\fglam\client\5.8.5.8-201408261438-
BETA-
     427
```

Starting the Foglight Agent Manager

```
C:\Quest\Foglight_Agent_Manager\bin>fglam -s
2011-11-22 14:25:17.000 INFO [native] Inter-launcher communications channel
   active at: C:\Quest\Foglight_Agent_Manager\state\default\
   .AgentManager-Host1.msg
2011-11-22 14:25:17.590 INFO Agent Manager: 5.6.2.2 (build 5622-20111115-1816-
   b29)
2011-11-22 14:25:22.448 ECHO <org.quartz.core.SchedulerSignalerImpl> INFO
   Initialized Scheduler Signaller of type: class
```

```
    org.quartz.core.SchedulerSignalerImpl
2011-11-22 14:25:24.406 ECHO <org.mortbay.log> INFO Logging to
    org.slf4j.impl.JCLLoggerAdapter(org.mortbay.log) via
    org.mortbay.log.Slf4jLog
2011-11-22 14:25:24.852 ECHO <org.mortbay.log> INFO jetty-6.1.26
2011-11-22 14:25:24.952 ECHO <org.mortbay.log> INFO Started
    SelectChannelConnector@127.0.0.1:63988
2011-11-22 14:25:24.953 INFO Started accepting HTTP connections at the local
    address 127.0.0.1:63988. These connections are only accessible from within
    this system.
2011-11-22 14:25:25.487 INFO Scanning for installed and recognized agent types
    in C:\Quest\Foglight_Agent_Manager\agents\HostAgents\6.0.0-6.0.0-
    20111116-1426-b43.2011-11-22 14:25:25.569 INFO Core startup complete
2011-11-22 14:25:26.136 ECHO <HostAgents/6.0.0/WindowsAgent/Monitor@Host1> INFO
    Created agent
2011-11-22 14:25:26.140 ECHO <HostAgents/6.0.0/WindowsAgent/Monitor@Host1> INFO
    Started data collection
2011-11-22 14:25:26.285 INFO Connected to upstream server at http://
    192.168.129.128:8080.
```

For information on how to deploy and activate a Foglight agent using the command-line interface, see Deploying Agent Packages on page 57 and Activating or Deactivating Agent Instances on page 63, respectively.

Stopping the Foglight Agent Manager

```
C:\Quest\Foglight_Agent_Manager\bin>fglam -q -w
2011-11-22 14:31:10.000 INFO [native] Shutdown request transmitted.
2011-11-22 14:31:10.000 INFO [native] Waiting for the AgentManager process to
    shutdown
2011-11-22 14:31:11.000 INFO [native] The AgentManager process has completed its
    shutdown sequence
```

Similarly to the previous example, if there are any active agents running on the Foglight Agent Manager, this command stops the agents' data collection and closes their Command Prompt windows (or terminal window).

Creating Agent Manager Support Bundles

support-bundle

The support-bundle command generates an Agent Manager support bundle file. A support bundle is a compressed file that contains diagnostic data, such as Agent Manager and individual agent log files.

When you create an Agent Manager bundle, Foglight generates a ZIP files in the *<foglight_home>/support/<user_name>* directory on the machine on which the Management Server is running. The support bundle file name uses the following syntax:

support_bundle_FglAM_<Agent Manager host>_<Management Server host>_<user name>-<yyyy-mm-dd>-<hh-mm-ss>**.zip**

For example:
```
support_bundle_FglAM_Host1.example.com_Host2.example.com_jsmith_2011-10-06_11-36-
39.zip
```

For details about the contents of this file, see the *Administration and Configuration Help*. For more information about *foglight_agent_mgr_home*, see Syntax Conventions on page 7.

Syntax

```
support_bundle
```

Options and arguments

None

Example

```
C:\Quest\Foglight_Agent_Manager\fglam\bin>support-bundle
.
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
     Attempting to obtain a support bundle from
     a running instance of the Foglight Agent Manager
-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=-=
.
5/13/21-07-29 22:08:48.000 INFO [native] Support bundle request sent.
```

Managing Agents, Cartridges, and Metrics

In addition to the Foglight Administration module that allows you to perform administrative tasks using the Administration dashboards, the Foglight Management Server offers a command-line interface that you can use to manage agents, cartridges and metrics. For example, you can use the command-line interface to deploy and create agents, install and activate cartridges, manage licenses, and do many other tasks.

The range and type of actions you can perform depends on the set of permissions defined by your Foglight user account.

- Understanding the Command-Line Syntax
- About Regular Expressions
- Getting Started with Administrative Commands
- Looking at Scope-Specific Commands
- Logging In and Setting the Scope
- Using a Properties File to Supply Connection Information
- Listing Agent and Agent Manager Instances
- Upgrading the Foglight Agent Manager
- Deploying Agent Packages
- Creating or Deleting Agent Instances
- Activating or Deactivating Agent Instances
- Starting or Stopping Data Collection
- Listing and Retrieving Log Files
- Listing and Assigning Blackout Schedules to Agent Instances
- Installing or Uninstalling Cartridges
- Enabling or Disabling Cartridges
- Listing Cartridges
- Managing Security Entities
- Managing Foglight Licenses
- Exporting or Importing a Monitoring Policy
- Exporting, Importing, or Listing UI Modules
- Exporting Metrics and Topology
- Looking Up Server Parameters
- Listing and Assigning Blackout Schedules to Topology Objects
- Listing Schedules
- Running Scripts

- Creating Server Support Bundles

Understanding the Command-Line Syntax

The command-line expression used to issue administrative commands on the Foglight Management Server is comprised of two commands that are separated by a colon ':' and appear in the following order:

- *fglcmd*. Logs into the Foglight Management Server and specifies the scope. It can contain two components:

  > **NOTE:** Only users with the Command Line Access privilege can use fglcmd. For more information about user privileges, refer to the Foglight Administration and Configuration Guide.

  - Log in information, `connection_options`, specifies the user name and password for the Foglight Management Server. It can also contain the machine name and port number. For complete information about the specific options that you can use to specify connection information on the command line, see Logging In and Setting the Scope on page 46.

    Another way of specifying connection options is using a properties file, `fglcmd.properties`. When configured, `fglcmd` uses the information in that file to retrieve connection parameters which eliminates the need to specify them on the command line each time you issue an `fglcmd` command. For more information, see Using a Properties File to Supply Connection Information on page 48.

  - The scope defines the type of commands and the target entity they are to be issued against. For example, the *cartridge* scope includes commands for installing and activating cartridges.The scope can be set to one of the following values:

Table 1. fglcmd command scope

Scope	Description
agent	Manages agents and agent managers
cartridge	Manages cartridges
license	Manages licenses
schedule	Lists schedules
script	Runs scripts
support	Creates support bundles
topology	Assigns blackout schedules
util	Contains utility commands

> **IMPORTANT:** The fglcmd part in the command-line expression that verifies the user and specifies the scope is required each time you issue commands for managing cartridges, agents, and metrics that are described in this chapter.

For complete information about *fglcmd*, see Logging In and Setting the Scope on page 46.

- *Scope-specific command*. Carries out an administrative task such as a cartridge deployment or license installation, and is compatible with the scope specified with `fglcmd`. For a complete list of Foglight commands and their respective scope, see Looking at Scope-Specific Commands on page 44.
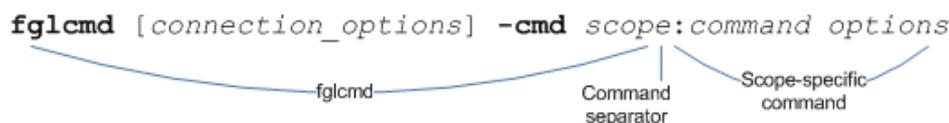
See also

About Regular Expressions

A regular expression describes one or more text strings using predefined syntactic elements. Regular expressions allow you to identify a list of text entries with a single text string. A number of `fglcmd` commands support regular expressions. This is useful in situations when you want to issue a single command against multiple objects. For example, if you need to generate a list of all Foglight Agent Manager instances running on a set of monitored hosts, you can use a regular expression to specify the names of those hosts in a single function call.

The backslash character '\' escapes a character or construct in order to process them as a literal sequence. For example, "\\" matches a single backslash while "\{" matches a left brace.

A regular expression must match the entire text string, not just a part of it. When using regular expressions to match an argument, you must specify an exact match or use wild cards. For example, if you have a host in your environment named `host1.example.com` that you want to match using a regular expression with the `clientname` argument, the outcomes of the regular expressions used as `clientname` arguments in the following commands are as follows:

Expression A

```
fglcmd -usr foglight -pwd foglight -port 8080 -cmd agent:create -clientname 'host1'
-type Process -name test -allclients -type Process -name test -force -regex
```

Outcome

```
Error : No clients are selected by current options
```

Expression B

```
fglcmd -usr foglight -pwd foglight -port 8080 -cmd agent:create-clientname 'host1.*'
-type Process -name test -allclients -type Process -name test -force -regex
```

Outcome

Agent created.

The following table lists the constructs that can be used in regular expressions and describes their usage.

Table 2. Regular expression syntax descriptions

Syntax	Description
Any Platform	
abc	The literal string contained within the construct. In `fglcmd`, using a literal string in a regular expression as a command argument and enabling the command to use regular expressions causes the command to return matches that contain that text string. For example, typing `mydomain.corp` as a regular expression finds matches that contain that text string.

Table 2. Regular expression syntax descriptions

Syntax	Description
****	A backslash character in regular expressions can have one of the following roles: • *Switch (or option)*. When it follows a command and is followed by an appropriate character or text string, it can provide additional input to the command. • *Escape character*. Signifies that the character or text string that follows should be interpreted as a literal character. For example, on the command line, the construct `rm *` deletes all files in the current directory while `rm \*` deletes only the file whose name is `*`. In regular expressions, the first backslash in a double-backslash construct acts as an escape character while the second is interpreted as a literal. For example, the construct `"\\(hello world\\)"` matches `"\(hello world\)"`.
\0n	A one-digit number with a value between zero '0' and seven '7'.
\0nn	A two-digit number with each digit having a value between zero '0' and seven '7'.
\0mnn	A three-digit number with the first digit having a value between zero '0' and three '3', and the other two digits with a value between zero '0' and seven '7'.
\xhh	A string containing a hexadecimal value of $0xhh$ where *h* is a hexadecimal digit.
\uhhhh	A string containing a hexadecimal value of $0xhhhh$ where *h* is a hexadecimal digit.
\t	The TAB character ('\u0009').
\n	The line feed character ('\u000A').
\r	The carriage return character ('\u000D').
\f	The form feed character ('\u000C').
\a	The alert (bell) character ('\u0007').
\e	The ESC character ('\u001B').
\cx	The CTRL character followed by a literal.
[…]	An OR expression. Brackets can be nested. Matches one of the characters within the brackets. For example, `[xyz]` matches `x`, `y`, or `z`.
[^…]	A negative OR expression. Matches any character that is not contained within the brackets. For example, `[xyz]` matches any character other than `x`, `y`, or `z`
-	A range. For example, `[a-d]` matches `a`, `b`, `c`, and `d`.
&&	A logical AND operator. For example, `[a-d]&&[m-p]` matches `a`, `b`, `c`, `d`, `m`, `n`, `o`, and `p`.

NOTE: The precedence of constructs used for combining the operators escape '\', grouping "[]", range '-'. union "[][]", or intersections "&&" (for example, in [a-z&&[aeiou]]) is done in the following order: escape, grouping, range, union, and intersection.

.	Any character
\d	Any decimal digit.
\D	Any character other than a decimal digit.
\s	A white space character such as a tab, line feed, blank space, or carriage return.
\S	Any character other than tab, line feed, blank space, or carriage return.
\w	Any lowercase or uppercase alphabetic, or a numeric character.
\W	Any character other than lowercase or uppercase alphabetic, or a numeric character.

Table 2. Regular expression syntax descriptions

Syntax	Description
?, ??, or ?+	When following a character, the construct implies that the preceding character can appear one or zero times. For example, each of the following constructs mean that the character X can appear once or not at all in the result: • `X?` • `X??` • `X?+`
*, *?, or *+	When following a character, the construct implies that the preceding character can appear zero or more times. For example, each of the following constructs mean that the character X can appear zero or more times in the result: • `X*` • `X*?` • `X*+`
+, +?, or ++	When following a character, the construct implies that the preceding character can appear one or more times. For example, each of the following constructs mean that the character X can appear one or more times in the result: • `X+` • `X+?` • `X++`
{n}, {n}?, or {n}+	When following a character, the construct implies that the preceding character can appear exactly *n* times. For example, each of the following constructs mean that the character X can appear exactly three times in the result: • `X{3}` • `X{3}?` • `X{3}+`
{n,}, {n,}?, or {n,}+	When following a character, the construct implies that the preceding character can appear at least *n* times. For example, each of the following constructs mean that the character X can appear at least five times in the result: • `X{5,}` • `X{5,}?` • `X{5,}+`
{n, m}, {n, m}?, or {n, m}+	When following a character, the construct implies that the preceding character can appear at least *n*, but no more than *m* times. For example, each of the following constructs mean that the character X can appear at least four, but no more than eight times in the result: • `X{4,8}` • `X{4,8}?` • `X{4,8}+`
NM	The construct implies that both characters appear in the given order: the first one (*N*) is followed by the second character (*M*) in the result, treating the two-character construct as a literal expression. For example, the expression XY returns XY as a match.
|	The logical OR operator. For example, the construct X|Y mean that either X or Y can appear in the result.

Table 2. Regular expression syntax descriptions

Syntax	Description
\Q	Quotes all characters in the expression until it reaches **\E**. For example, the construct `\Qabc\E` has the same meaning as `"abc"`.
\E	Ends the quoting started by \Q.
Groups	
()	Parentheses are used to create capturing groups. A capturing group in a text pattern is used to match sub-strings in expressions. For example, in the construct `X(Y*)Z`, the capturing group `(Y*)` matches both `Y` and `YY` from the input, returning both `XYZ` and `XYYZ` as the result of the expression. Capturing groups can be nested and numbered using their opening parentheses from left to right. For example, in the construct `((X(Y))(Z))`, the groups are numbered as follows: • `((X(Y))(Z))`: group 1 • `(X(Y))`: group 2 • `(Y)`: group 3 • `(Z)`: group 4
\n	Following a series of capturing groups, it acts as a back reference to match of the *n*th group. For example, the expression ([a-d])X\1X\1 has only one capturing group whose number is one '1'. It returns the following matches: • `aXaXa` • `bXbXb` • `cXcXc` • `cXcXc`
(?:N)	Indicates that *N* is a non-capturing group in a construct. For example, in the construct `(X(?:Y))(Z)`, the group `(?:Y)` is not considered as a capturing group. The groups in the above construct are numbered as follows. • `(X(?:Y))(Z)`: group 1 • `(X(?:Y))`: group 2 • `(Z)`: group 3 For information about capturing groups and their syntax, see () on page 40.
(?=X)	Checks if the preceding character is followed by *X* in a text string, without making *X* a part of the search result. For example, when the construct `H(?=e)` is matched against `Hello world`, it returns the `H` in the string without making the `e` that follows it a part of the result.
(?!X)	Checks if the preceding character is preceded by *X* in a text string, without making *X* a part of the search result. For example, when the construct `e(?!H)` is matched against `Hello world`, it returns the `e` in the string without making the `H` that precedes it a part of the result.
(?<=X)	Checks if the following character is followed by *X* in a text string, without making *X* a part of the search result. For example, when the construct `(?<=w)o` is matched against `Hello world`, it returns the `o` in world, without making the `w` a part of the result, but not the `o` in `Hello`. Returns *A* via zero-width positive look behind.
(?<!X)	Checks if the following character is preceded by *X*, without making *X* a part of the search result. For example, when the construct `(?<!o)w` is matched against `Hello world`, it returns the `w` in world, without making the `w` a part of the result.

Table 2. Regular expression syntax descriptions

Syntax	Description
Boundary matching	
^	The beginning of a line.
$	The end of a line.
\b	A word boundary. Used as a delimiter, it implies that the construct between the delimiters should be matched only in those text strings that contain alpha-numeric characters and are delimited by non-word character such as spaces or punctuation marks. For example, the construct `\bdog\b` finds one match of `dog` in the string `My dog is black`, and no matches in `My dogs are black`.
\B	A non-word boundary. Used as a delimiter, \B is the negated version of \b. For example, the construct `\Bdog\B` finds a match of `dog` in the string `My dog is black`, and one in `My dogs are black`.
\A	The beginning of the input. It has the same functionality as ^, with the exception that it ignores any new line characters.
\G	The end of the previous match.
\Z	The end of the input string.It has the same functionality as $, with the exception that it ignores any final terminators.
\z	The end of the input string.It has the same functionality as $, with the exception that it ignores any line terminators.
Unicode blocks	
\p{Lu}	Any uppercase alphanumeric character.
[\p{L}&&[^\p{Lu}]]	Any lowercase alphanumeric character.
POSIX (US-ASCII only)	
\p{Upper}	Any upper-case alphabetic character.
\p{ASCII}	Any ASCII character.
\p{Alpha}	Any lower-case or upper-case alphabetic character.
\p{Digit}	Any decimal digit.
\p{Alnum}	Any lower-case or upper-case alphabetic character, or a numeric character.
\p{Punct}	One of the following punctuation characters: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~
\p{Graph}	Any alphabetic (lowercase or uppercase), numeric, or punctuation character.
\p{Print}	Any printable alphabetic (lowercase or uppercase), numeric, or punctuation character.
\p{Blank}	A blank space or a TAB character.
\p{Cntrl}	A CTRL character.
\p{XDigit}	A hexadecimal digit.
\p{Space}	A white space character such as a tab, line feed, blank space, or carriage return.

> **i** | **NOTE:** Regular expressions in Foglight follow the Java[TM] guidelines for regular expressions. For complete information, you can refer to JDK documentation.

See also

Getting Started with Administrative Commands

You can issue the administrative commands on the computer running the Foglight Management Server, or on a remote computer in your network. Your Foglight installation includes a compressed file, *fglcmd.zip*, that contains all of the administrative commands. To get started, make sure you have the Foglight Management Server and Foglight Agent Manager running, and, if you are planning to access the administrative interface remotely, copy and extract the contents of the compressed file to your remote computer.

To get started with administrative commands:

> **i** | **NOTE:** This procedure continues from Getting Started with Foglight Commands on page 11.

1 If you want to use the *fglcmd* interface on a remote computer in your monitoring network, complete the following steps:

    a Copy the *<foglight_home>/tools/fglcmd.zip* file from the computer that has a running instance of the Foglight Management Server to the remote computer, where *foglight_home* refers to the installation directory of the Foglight Management Server.

    b Extract the contents of the *fglcmd.zip* file to the remote computer.

2 Specify the directory that contains the Foglight Management Server commands, either *<foglight_home>/bin* or the directory on the remote computer (see Step 1), by completing one of the following steps.

    ▪ If you want to use the command prompt, open a Command Prompt window (Windows$^{®}$) or a terminal window (UNIX$^{®}$ or Linux$^{®}$) and navigate to the appropriate directory.

    or

    ▪ If you want to use a Foglight Management Server command in a script, ensure that your script references the appropriate directory.

    For more information about *foglight_home*, see Syntax Conventions on page 7.

3 **Foglight Management Server enables only the HTTPS port for remote access, and installs a self-signed certificate**. The security certificate allows the server to communicate with a remote machine through the HTTPS protocol. If you intend to use the administrative commands either on a remote computer, or on a Foglight Management Server computer through the HTTPS protocol, you need to export the security certificate from the Foglight Management Server, and import it into your JRE environment, as described below.

> **i** | **IMPORTANT:** The commands below follow the generic syntax in which forward slashes '/' are used to separate directories. On Windows platforms, instead of forward slashes '/', use back slashes '\' as directory separators in file paths. For more information about the syntax conventions used in this manual, see Syntax Conventions on page 7.

    a Export the security certificate from the Foglight Management Server.

    On the computer on which a Foglight Management Server is running, open a Command Prompt window (Windows) or a terminal window (UNIX$^{®}$ or Linux$^{®}$) and navigate to *<foglight_home>/jre/bin*, followed by issuing the following command:

```
keytool -exportcert -keystore foglight_home/config/tomcat.keystore -alias tomcat -file filename -storepass nitrogen
```

    Where *filename* is the name of the file in which the command stores the security certificate.

b   Import the security certificate into your environment, either on the remote computer, or on the computer on which the Foglight Management Server is running.

On the computer on which you intend to use the administrative commands, open a Command Prompt window (Windows) or a terminal window (UNIX® or Linux®) and navigate to *<jre_home>/bin*, followed by issuing the following command:

```
keytool -importcert -file filename -alias foglightcert -keystore
../lib/security/cacerts -storepass changeit
```

Where *filename* is the name of the file to which you exported the security certificate in Step a.

> **i** | **IMPORTANT:** The default keystore password for jre/lib/security/cacerts is changeit.

4   Issue one or more administrative commands using command prompt or a script. From here, you can issue any `fglcmd` commands described in the following sections, as required.

See also

Running Administrative Commands: Example

In this example, you will issue an administrative command that lists the agent packages that can be deployed to the Foglight Agent Manager using the `packages` command. The syntax of the `packages` command is as follows:

```
fglcmd connection_options -cmd agent:packages options
```

For more information about the command syntax, see Managing Agents, Cartridges, and Metrics on page 35.

In this example, you will issue the packages administrative command (see page 52) to list all Foglight Agent Manager instances and the agent packages that can be deployed to each Agent Manager instance.

To list agent packages:

> **i** | **NOTE:** This procedure continues from "Getting Started with Foglight Commands on page 11.

- In the Command Prompt window (Windows®) or the terminal window (UNIX® or Linux®), type the following command:

```
fglcmd -usr foglight -pwd foglight -cmd agent:packages -allclients
```

A list of all available agent packages appears in the Command Prompt or terminal window. A sample of that output is similar to the following listing:

```
Client ID: Host1-2d2edfcc#be05acc0-6179-4878-a274-afcd5075260a
   Agent Package ID: HostAgents-6.0.0-Host Agents
   Agent Package Cartridge Name: HostAgents
   Agent Package Cartridge Version: 6.0.0
   Agent Package Version: 6.0.0
   ---------------------------------
Client ID: Host2.example.com#6290d00c-8fd6-42a3-bc2a-63b0d5263645
   Agent Package ID: HostAgents-6.0.0-Host Agents
   Agent Package Cartridge Name: HostAgents
   Agent Package Cartridge Version: 6.0.0
```

```
Agent Package Version: 6.0.0
---------------------------------
```

See also

- Managing Agents, Cartridges, and Metrics on page 35
- About Regular Expressions on page 37
- Getting Started with Administrative Commands on page 42
- Looking at Scope-Specific Commands on page 44

Looking at Scope-Specific Commands

Foglight includes a selection of commands that allow one to perform administrative tasks using a command-line interface, such as installing and deploying agents, adding or removing licenses, assigning schedules, or exporting various types of metrics. The table below lists Foglight commands in alphabetical order and identifies their scope. For more information about command scope, see Managing Agents, Cartridges, and Metrics on page 35.

Table 3. Scope-specific commands

Scope		Command	
Name	**Description**	**Name**	**Description**
agent	Manages agents and agent managers	**activate**	Activates one or more Foglight agent instances
		clients	Shows a list of Foglight agent manager instances
		clientupgrade	Upgrades one or more remote instances of the Foglight Agent Manager
		create	Creates one or more Foglight agent instances
		deactivate	Deactivates one or more Foglight agents
		delete	Deletes one or more Foglight agent instances
		deploy	Deploys one or more Foglight agent packages
		getlog	Retrieves a copy of an agent log file
		list	Shows a list of Foglight agent instances
		logs	Shows a list of agent log files
		packages	Shows a list of available agent packages
		setschedule	Assigns a blackout schedule to one or more Foglight agent instances
		showschedule	Shows the blackout schedule assigned to one or more Foglight agent instances.
		start	Starts the data collection for one or more Foglight agent instances
		stop	Stops the data collection for one or more Foglight agent instances
		types	Shows a list of Foglight agent types sorted by agent manager ID
cartridge	Manages cartridges	**disable**	Deactivates a Foglight cartridge
		enable	Activates an installed Foglight cartridge
		install	Installs a cartridge on the Foglight Management Server
		list	Lists all installed Foglight cartridges
		uninstall	Uninstalls a Foglight cartridge

Table 3. Scope-specific commands

Scope		Command	
Name	**Description**	**Name**	**Description**
license	Manages licenses	**import**	Installs a Foglight license
		list	Shows a list of installed Foglight licenses
		remove	Removes a Foglight license using the license serial number
schedule	Lists schedules	**list**	Shows a list of all Foglight schedules
script	Runs scripts	**run**	Runs a script
security	Manages users and groups	**assigngroup**	Adds or removes a user from a group
		assignrole	Adds or removes a role from a group
		creategroup	Creates a group
		createuser	Creates a user
		deletegroup	Deletes a group
		deleteuser	Deletes a user
		exportldapcfg	Exports the LDAP settings to an XML file
		importldapcfg	Imports the LDAP settings from an XML file
		list	Lists users, groups, and roles
support	Creates support bundles	**bundle**	Generates a support bundle file
topology	Lists and assigns blackout schedules	**blackoutobject**	Assigns a blackout schedule
		blackouts	Lists the blackout schedules assigned to topology objects
util	Contains utility commands	**configexport**	Exports the monitoring policy to an XML file
		configimport	Imports the monitoring policy from an XML file
		env	Outputs the values of server configuration parameters
		metricexport	Exports metric observations to a file using a metric query
		topologyexport	Exports the value of one or more properties of a topology object to an XML file
		uiexport	Exports a UI module to a ZIP file
		uiimport	Imports a UI module from a ZIP file
		uilist	Shows a list of deployed UI modules

Listing Administrative Commands

The *fglcmd* command includes an option that you can use to list all of the administrative commands and their descriptions, grouped by their respective scope.

To list administrative commands:

> **i** | **NOTE:** This procedure continues from Getting Started with Foglight Commands on page 11.

- At the command prompt, use the following syntax to list Foglight commands:

  `fglcmd` *connection_options* `-cmd commands`

  For complete information about the *fglcmd* command syntax, see Syntax in fglcmd on page 46.

For example:

```
fglcmd -usr foglight -pwd foglight -cmd commands
```

A list of Foglight commands appears in the Command Prompt window (Windows®) or the terminal window (UNIX® or Linux®), grouped by the scope. Here is a portion of that output:

```
agent:
   activate      Activates an agent.
   clients       Shows a list of known clients.
   clientupgrade Upgrades a remote client.
   create        Creates an instance of an agent.
   deactivate    Deactivates an agent.
```

For complete information about **fglcmd**, see Logging In and Setting the Scope on page 46.

See also

- Managing Agents, Cartridges, and Metrics on page 35
- Getting Started with Administrative Commands on page 42
- Running Administrative Commands: Example on page 43

Logging In and Setting the Scope

fglcmd

The `fglcmd` command connects to the Foglight Management Server and sets the scope for the administrative command.

Syntax

```
fglcmd {(-usr user_name -pwd password)|-authtoken token} [-debug]
   [-port server_port] [-srv {server_name|server_IP_address}] [-ssl]
   [-proxy proxy_uri] -cmd {scope:command command_options|commands}
```

> **i** | **NOTE:** The `usr`, `pwd`, `debug`, `port`, `srv`, and `ssl` options are used to specify connection information. Another way of specifying connection options is using a properties file, *fglcmd.properties*. When configured, `fglcmd` uses the information in that file to retrieve connection parameters which eliminates the need to specify them on the command line each time you issue an fglcmd command. For more information, see Using a Properties File to Supply Connection Information on page 48.

Table 4. Options and arguments

Option	Argument	Description	
authtoken	*token*	Specifies the auth-token.	
cmd	*scope*	**agent**	Manages agents and agent managers.
		cartridge	Manages cartridges.
		commands	Lists administrative commands.
		license	Manages licenses.
		schedule	Lists schedules.
		script	Runs scripts.
		security	Manages security entities.
		support	Creates support bundles.
		topology	Assigns blackout schedules.
		util	Contains utility commands.
debug	None	Turns the debugging on.	
port	*server_port*	Specifies the server's port number. The default port number is `8080` (HTTP) or `8443` (HTTPS). For more information on default port assignments, see the *Administration and Configuration Help*.	
proxy	*proxy-uri*	Specifies the URI of proxy server. Three types of proxy protocols with or without authentication are supported, including HTTP, HTTPs, and SOCKS. The generic syntax is: `{http|https|socks}://[username:password@]{proxy_server_hostname|IP}:port`	
pwd	*password*	Specifies the password of Foglight user. **NOTE:** If the password argument is not provided, `fglcmd` prompts for the password after a command is issued.	
srv	*server_name*	Specifies the server's name.	
	server_IP_address	Specifies the server's IP address.	
ssl	None	Connects to the Foglight Management Server through a secure HTTP connection (HTTPS). Unless this option is specified, **fglcmd** uses an HTTP connection. For information about default port numbers, see port on page 47.	
usr	*user_name*	Specifies the user name for logging into Foglight.	

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:clients -host
.*.corp -regex
Client ID: host1.example.com#32b1c51d-6208-435c-bdad-
   2a8916ef3a89
Client Name: host1.example.com
Adapter ID: FglAM
Client Version: 6.0.0 (6.0.0-5/13/210811-0230)
Host Name: host1.example.com
OS: windows 5.1 (ia32)
```

See also

- Managing Agents, Cartridges, and Metrics on page 35
- Looking at Scope-Specific Commands on page 44
- Running Administrative Commands: Example on page 43

Using a Properties File to Supply Connection Information

Foglight connection information, such as the user name and password, can be supplied on the command line, using a specific set of command-line options, or through a properties file. Specifying connection options in the properties file eliminates the need to type them each time you issue an `fglcmd` command.

This section describes the required structure and location of that file. For information on specifying connection options on the command line, see Logging In and Setting the Scope on page 46.

File name and location

Unix

~/fglcmd.properties

Windows

C:\Documents and Settings\<user_name>\fglcmd.properties

Syntax

usr=*user_name*

pwd=*password*

> **i** | **TIP:** If you specify the password in this file, it is recommended to read-protect the properties file. For example, to read-protect a file on Unix, use the `-R world` option; on Windows, files can be read-protected by associating read permissions with particular users or groups.

```
[srv={server_name|server_IP_address}]
[port=server_port]
[debug={true|false}]
[ssl={true|false}]
```

> **i** | **IMPORTANT:** The `cmd` option, used to specify a specific `fglcmd` command, cannot be set through the properties file. This option can only be used when issuing `fglcmd` commands.

Table 5. Options and arguments

Option	Argument	Description
debug	**true** or **false**	Turns the debugging on (`true`) or off (`false`).
port	*server_port*	Specifies the server's port number. The default port number is `8080` (HTTP) or `8443` (HTTPS). For more information on default port assignments, see the *Administration and Configuration Help*.
pwd	*password*	Specifies the password of the Foglight user. **NOTE:** If the password argument is not provided, `fglcmd` prompts for the password after a command is issued.
srv	*server_name*	Specifies the server's name.
	server_IP_address	Specifies the server's IP address.

Table 5. Options and arguments

Option	Argument	Description
ssl	None	Connects to the Foglight Management Server through a secure HTTP connection (HTTPS). Unless this option is specified, **fglcmd** uses an HTTP connection. For information about default port numbers, see port on page 47.
usr	*user_name*	Specifies the user name for logging into Foglight.

Example

File name and location

C:\Documents and Settings\<user_name>\fglcmd.properties

fglcmd.properties contents

```
usr=bsmith
pwd=secret
srv=production
port=8080
debug=true
ssl=true
```

Command-line output

```
C:\Quest\Foglight\bin>fglcmd -cmd agent:clients -host .*.corp -regex
Client ID: host1.example.com#e4ef68f8-08ec-4a4f-9f43-63f2be956bae
Client Name: host1.example.com
Adapter ID: FglAM
Client Version: 6.0.0 (6.0.0-20090513-0250)
Host Name: host1.example.com
OS: windows 5.1 (ia32)
```

See also

- Managing Agents, Cartridges, and Metrics on page 35
- Looking at Scope-Specific Commands on page 44
- Running Administrative Commands: Example on page 43
- Logging In and Setting the Scope on page 46

Listing Agent and Agent Manager Instances

clients

The `clients` command shows a list of Foglight Agent Manager instances.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:clients [-regex] [-host host_name]
    [-clientname display_name] [-clientid client_ID] [-clientbuild build_ID]
    [-clientversion version] [-allclients] [-upgradable {true|false}]
```

Table 6. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be listed.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more instances of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies the name of one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
host	*host_name*	An explicit text string or a regular expression that specifies the name of one or more hosts that are running an instance of the Foglight Agent Manager.
regex	None	Interprets the *client_ID*, *display_name*, *host_name,* or *upgradable* arguments as regular expressions. For a sample regular expression used to specify a host name, see the Example.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:clients -host
".*.corp" -regex
Client ID: host1.example.com#32b1c51d-6208-435c-bdad-2a8916ef3a89
Client Name: host1.example.com
Adapter ID: FglAM
Client Version: 6.0.0 (6.0.0-5/13/210811-0230)
Host Name: host1.example.com
OS: windows 5.1 (ia32)
```

For complete information about the regular expression syntax, see About Regular Expressions on page 37.

See also

- list on page 51

list

The `list` command shows a list of Foglight agent instances.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:list [-agentid agent_ID] [-regex]
    [-adapter adapter_ID] [-host host_name] [-deletable {true|false}]
    [-name agent_name] [-namespace agent_namespace]
    [-datacollection {true|false}] [-all] [-type agent_type]
    [-activatable {true|false}] [-active {true|false}]
```

Table 7. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to list agent instances that can (**true**) or cannot be activated (**false**).
active	**true** or **false**	Indicates whether to list agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight™ adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be listed.
all	None	Indicates that all agent instances should be listed.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to list Foglight agent instances for which the data collection is (**true**) or is not enabled (**false**).
deletable	**true** or **false**	Indicates whether to list Foglight agent instances that can (**true**) or cannot be deleted (**false**).
host	*host_name*	An explicit text string or a regular expression that specifies the name of one or more hosts that are running an instance of a Foglight agent.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances that are to be listed.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.

Table 7. Options and arguments

Option	Argument	Description
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type,* or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Managing the Infrastructure Cartridge User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight
   -cmd agent:list -host Host1 -active true

Host: Host1
ID: 1
Name: Monitor@Host1
Type: WindowsAgent
Version: 6.0.0 (Build: 6.0.0-20111116-1426-b43)
Adapter ID: FglAM
Status: Active/Collecting data
Health State: OK
----------------------------------
```

See also

- clients on page 49
- packages on page 52
- types on page 54

packages

The `packages` command shows a list of available agent packages.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:packages [-packageosversion OS_version]
   [-regex] [-host host_name] [-clientname display_name]
   [-packageversion version] [-arch architecture] [-packageid "pkg_ID"]
```

```
[-allclients] [-allpackages] [-allinstallers] [-osversion OS_version]
[-packagearch OS_architecture] [-clientid client_ID] [-clientbuild build_ID]
[-clientversion client_version] [-packageos OS_name] [-osname OS_name]
[-installername name] [-upgradable {true|false}] [-disable_platform_check]
```

Table 8. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
allinstallers	None	Indicates that all agent installers should be selected.
allpackages	None	Indicates that all agent packages should be listed.
arch	*architecture*	An explicit text string or a regular expression that specifies the platform architecture.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more instances of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*client_version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
disable_platform_check	None	Disables automatic filtering of agent packages, based on platform information provided by agent installers.
host	host_name	An explicit text string or a regular expression that specifies one or more host names.
installername	*name*	An explicit text string or a regular expression that selects one or more agent installers based on the specified name.
osname	*OS_name*	An explicit text string or a regular expression that specifies the name of one or more operating systems with which the agent packages are compatible.
osversion	*OS_version*	An explicit text string or a regular expression that specifies the version of one or more operating systems with which the agent packages are compatible.
packagearch	*OS_architecture*	An explicit text string or a regular expression that specifies one or more operating system architectures with which the agent packages are compatible.
packageid	*pkg_ID*	An explicit text string or a regular expression that identifies one or more packages. **NOTE:** This value must be enclosed in quotation marks, as indicated in the command syntax.
packageos	*pkg_OS_name*	An explicit text string or a regular expression that specifies the name of one or more operating systems of the agent packages that are to be selected.

Table 8. Options and arguments

Option	Argument	Description
packageosversion	*pkg_OS_version*	An explicit text string or a regular expression that specifies one or more operating system versions of the agent packages that are to be selected.
packageversion	*version*	An explicit text string or a regular expression that specifies one or more package versions.
regex	None	Interprets the *architecture*, *build_ID, client_ID, client_version, display_name*, *host_name*, *name*, *OS_architecture*, *OS_name*, *OS_version*, *pkg_ID*, *pkg_OS_name*, *pkg_OS_version*, *version*, or *upgradable* arguments as regular expressions. For a sample regular expression, see Example on page 50.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd agent:packages -allclients**

```
Client ID: Host1#be05acc0-6179-4878-a274-afcd5075260a
   Agent Package ID: HostAgents-6.0.0-Host Agents
   Agent Package Cartridge Name: HostAgents
   Agent Package Cartridge Version: 6.0.0
   Agent Package Version: 6.0.0
-----------------------------------
Client ID: Host3.example.com#6290d00c-8fd6-42a3-bc2a-63b0d5263645
   Agent Package ID: HostAgents-6.0.0-Host Agents
   Agent Package Cartridge Name: HostAgents
   Agent Package Cartridge Version: 6.0.0
   Agent Package Version: 6.0.0
-----------------------------------
```

See also

- clients on page 49
- list on page 51
- types on page 54

types

The **types** command shows a list of Foglight agent types that are available for deployment to individual Agent Managers, sorted by the Agent Manager ID.

Scope

`agent`

Syntax

fglcmd *connection_options* **-cmd agent:types** [**-regex**] [**-host** *host_name*]

```
[-clientname display_name] [-clientid client_ID] [-clientbuild build_ID]
[-clientversion version] [-allclients] [-upgradable {true|false}]
```

Table 9. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	client_ID	An explicit text string or a regular expression that identifies one or more instances of the Foglight Agent Manager.
clientname	display_name	An explicit text string or a regular expression that specifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
host	host_name	An explicit text string or a regular expression that specifies one or more host names.
regex	None	Interprets the *build_ID,* client_ID, display_name, host_name, *version,* or *upgradable* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:types -
clientname host1.example.com

Client ID: Host1#be05acc0-6179-4878-a274-afcd5075260a
Client Name: Host1
Agent Types:
  UnixAgent
  WindowsAgent
---------------------------------
```

See also

- clients on page 49
- list on page 51
- packages on page 52

Upgrading the Foglight Agent Manager

clientupgrade

The `clientupgrade` command can be used to upgrade one or more remote instances of the Foglight Agent Manager.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:clientupgrade {-clientname display_name|-
    clientid client_ID|-clientbuild build_ID|-clientversion version|-allclients}
    [-host host_name] {{-to version [-tobuild build_id] [-regex]
    [-timeout seconds] [-force]}|-list} [-upgradable {true|false}]}|
```

Table 10. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies the name of one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
force	None	In cases where multiple agents are selected, it indicates that the command should be performed against all selected agent installers and packages.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
list	None	Lists available upgrade versions.
regex	None	Interprets the *architecture*, *build_ID, client_ID*, *display_name*, *host_name*, *name*, *OS_architecture*, *OS_name*, *OS_version*, *package_ID*, *pkg_OS_name*, *pkg_OS_version*, *version,* or *upgradable* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.

Table 10. Options and arguments

Option	Argument	Description
timeout	*seconds*	Specifies the maximum time in seconds for the timeout period. If the timeout i not specified or is set to zero '0', the command waits for the upgrade to complete.
tobuild	build_ID	Specifies the build number to which the selected instances of the Foglight Agent Manager should be upgraded.
toversion	version	Specifies the version to which the selected instances of the Foglight Agent Manager should be upgraded.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:clientupgrade -
list -allclients -upgradable true
Client ID: Host1#be05acc0-6179-4878-a274-afcd5075260a
Client Name: Host1
Client Version: 5.6.2.2 (5622-20111116-1616-b30)
Upgradable: true
No upgrades are available for this client at this moment

Client ID: Host2.example.com#6290d00c-8fd6-42a3-bc2a-63b0d5263645
Client Name: Host2.example.com
Client Version: 5.6.2.2 (5622-20111115-1816-b29)
Upgradable: true
No upgrades are available for this client at this moment
```

Deploying Agent Packages

deploy

The `deploy` command deploys one or more Foglight agent packages.

In most cases, Foglight cartridges include one or more agent packages that are used to collect metrics from monitored hosts. You can deploy an agent package after installing the cartridge that includes that agent package on the Foglight Management Server. For information on how to install a cartridge using the `fglcmd` interface, see install on page 78; to find out how to list available agent packages using `fglcmd`, see packages on page 52.

Scope

agent

Syntax

```
fglcmd connection_options -cmd agent:deploy {-packageversion version|-packageid
  "package_ID"|-allpackages|-allinstallers} {-host host_name|-clientname
  display_name|-clientid client_ID|-clientbuild build_ID|-clientversion
  version|-allclients} [-packageos pkg_OS_name]
  [-packageosversion pkg_OS_version] [-packagearch OS_architecture]
  [-osversion OS_version] [-osname OS_name] [-arch architecture]
```

```
[-installerid agent_installer_ID] [-installername agent_installer_name]
[-regex] [-test] [-disable_platform_check] [-force]
[-upgradable {true|false}]
```

Table 11. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
allinstallers	None	Indicates that all agent installers should be selected.
allpackages	None	Indicates that all agent packages should be deployed.
arch	*architecture*	An explicit text string or a regular expression that specifies one or more platform architectures.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies the name of one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
disable_platform_check	None	Disables automatic filtering of agent packages based on platform information provided by agent installers.
force	None	In cases where multiple agents are selected, it indicates that the command should be performed against all selected agent installers and packages.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
installerid	*agent_installer_ID*	Selects the agent installer with the specified ID.
installername	*name*	An explicit text string or a regular expression that selects one or more agent installers based on the specified name.
osname	*OS_name*	An explicit text string or a regular expression that specifies the name of one or more operating systems with which the agent packages are compatible
osversion	*OS_version*	An explicit text string or a regular expression that specifies one or more operating system versions the agent packages are compatible with.
packagearch	*OS_architecture*	An explicit text string or a regular expression that specifies one or more operating system architectures the agent packages are compatible with.

Table 11. Options and arguments

Option	Argument	Description
packageid	*package_ID*	An explicit text string or a regular expression that identifies one or more agent packages that are to be deployed. **NOTE:** This value must be enclosed in quotation marks, as indicated in the command syntax.
packageos	*pkg_OS_name*	An explicit text string or a regular expression that specifies the name of one or more operating systems of the agent packages that are to be deployed.
packageosversion	*pkg_OS_version*	An explicit text string or a regular expression that specifies one or more operating system versions of the agent packages that are to be deployed.
packageversion	*version*	An explicit text string or a regular expression that specifies one or more versions of the agent packages that are to be deployed.
regex	None	Interprets the *architecture*, *build_ID*, *client_ID*, *display_name*, *host_name*, *name*, *OS_architecture*, *OS_name*, *OS_version*, *package_ID*, *pkg_OS_name*, *pkg_OS_version*, *version*, or *upgradable arguments* as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
test	None	Issues the command in test mode without deploying any agents.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:deploy -host
Host1 -clientid Host1#be05acc0-6179-4878-a274-afcd5075260a -packageid HostAgents-
6.0.0-Host Agents
```
Successfully installed agent package "HostAgents-6.0.0-Host Agents" on
"Host1#be05acc0-6179-4878-a274-afcd5075260a#FglAM:Host1/be05acc0-6179-4878-
a274-afcd5075260a/installer".

See also

- create on page 60
- delete on page 61
- activate on page 63
- deactivate on page 66
- start on page 68
- stop on page 69

Creating or Deleting Agent Instances

create

The `create` command produces one or more Foglight agent instances.

You can create an agent instance after deploying the agent package on the Foglight Management Server. To create an agent instance that collects data, you first create the agent instance, activate it, and then start the data collection for that agent instance. Similarly, when you want to remove an agent instance, use a reverse order of actions: first stop the agent instance, deactivate it, and then delete it.

For information on how to deploy an agent package using the `fglcmd` interface, see deploy on page 57; for information on how to delete agent instances, see delete on page 61.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:create -name agent_name -type agent_type
   {-clientname display_name|-clientid client_ID|-clientbuild build_ID|
   -clientversion version|-allclients} [-host host_name] [-force] [-nowait]
   [-regex] [-upgradable {true|false}]
```

Table 12. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies one or more monitored hosts on which the Foglight Agent Manager is running.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be created.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies one or more agent names.
nowait	None	Prevents the command from waiting for the selected agents to finish processing before creating the selected Foglight agent instances.

Table 12. Options and arguments

Option	Argument	Description
regex	None	Interprets the *build_ID., client_ID*, *display_name*, *host_name*, *version,* or *upgradable* arguments as regular expressions.
		For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types.
		The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types:
		• WindowsAgent
		• UnixAgent
		NOTE: For more information about these agents, see *Foglight for Infrastructure User and Reference Guide*.
upgradable	**true** or **false**	Indicates whether one or more Foglight
		Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

Using a regular expression to select hosts

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:create -type
Windows_System -host .*.corp -name MyAgentX -regex -force
```

  i | **NOTE:** If successful, this command does not generate any output.

Using a literal value to select a Foglight Agent Manager instance

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:create -type
WindowsAgent -name Agent5 -clientname Host1
```

  i | **NOTE:** If successful, this command does not generate any output.

See also

delete

The **delete** command removes one or more Foglight agent instances.

To delete an agent instance, a typical flow of actions requires you to first stop the agent's data collection, deactivate it, and then delete the instance. However, the delete command includes options and arguments, such

as `-active true` and `-datacollection true`, that let you override that flow and delete active agent instances that are collecting data.

For information on how to deactivate an agent instance using the `fglcmd` interface, see deactivate on page 66.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:delete {-agentid agent_ID|
   -name agent_name|-namespace agent_namespace|-all} [-type agent_type]
   [-host host_name] [-adapter adapter_ID] [-deletable {true|false}]
   [-datacollection {true|false}] [-activatable {true|false}]
   [-active {true|false}] [-regex] [-force] [-nowait]
```

Table 13. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to delete agent instances that can (**true**) or cannot be activated (**false**).
active	**true** or **false**	Indicates whether to delete agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be deleted.
all	None	Indicates that all agent instances should be deleted.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to delete Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to delete Foglight agent instances that can (**true**) or cannot be deleted (**false**).
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be deleted.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more Foglight agents that are to be deleted.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
nowait	None	Prevents the command from waiting for the selected agents to finish processing before deleting the selected Foglight agent instances.

Table 13. Options and arguments

Option	Argument	Description
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace, agent_type*, or *host_name* arguments as regular expressions.
		For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types.
		The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types:
		• WindowsAgent
		• UnixAgent
		For more information about these agents, see the *Managing the Infrastructure Cartridge User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:delete -name
Agent4
```

ℹ | **NOTE:** If successful, this command does not generate any output.

See also

- deploy on page 57
- create on page 60
- activate on page 63
- deactivate on page 66
- start on page 68
- stop on page 69

Activating or Deactivating Agent Instances

activate

The `activate` command enables one or more Foglight agent instances.

To create an agent instance that collects data, you first create the agent instance, activate it, and then start the data collection for that agent instance. Similarly, when you want to remove an agent instance, use a reverse order of actions: first stop the agent instance, deactivate it, and then delete it.

i | **TIP:** Some Foglight agents, such as the AIX® and HP® system agents that come included with the OS Cartridge, require root-level privileges to run on UNIX®. This can be accomplished by making sure that the user account used to install the Foglight Agent Manager on the monitored host does not appear in the sudo configuration file. However, if the Agent Manager does not have root-level privileges, the agents that require those privileges fall back to normal, non-root launch. Starting those agents without root privileges does not prevent them from activating and collecting data. For more information about installing the Foglight Agent Manager with root privileges, see the *Installation and Configuration Guide*. For information about the OS Cartridge, refer to the cartridge documentation.

For information on how to create an agent instance, see create on page 60; to find out how to deactivate an agent, see deactivate on page 66.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:activate
    {-agentid agent_ID|-name agent_name|-namespace agent_namespace|-all}
    [-host host_name] [-adapter adapter_ID] [-type agent_type]
    [-deletable {true|false}] [-datacollection {true|false}]
    [-activatable {true|false}] [-active {true|false}] [-regex] [-force]
    [-nowait]
```

Table 14. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to activate agent instances that can (**true**) or cannot be activated (**false**).
active	**true** or **false**	Indicates whether to activate agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be activated.
all	None	Indicates that all agent instances should be activated.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to activate Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to activate Foglight agent instances that can (**true**) or cannot be deleted (**false**).
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be activated.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more Foglight agents that are to be activated.

Table 14. Options and arguments

Option	Argument	Description
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
nowait	None	Indicates that the command should not wait for the selected agents to finish processing before activating the selected Foglight agent instances.
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions.
		For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types.
		The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types:
		• WindowsAgent
		• UnixAgent
		For more information about these agents, see the *Managing the Infrastructure Cartridge User and Reference Guide*.

Example

The first command in this example, shown in Listing agents on page 65, lists agent IDs using the `agent:list` command with the `all` option set. For more information on how to list agent instances, see list on page 51.

The second command, shown in Activating an agent instance on page 65, uses an ID of an inactive agent from the output of the first command to activate an agent.

Listing agents

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:list -all

Host: Host1
ID: 1
Name: Monitor@Host1
Type: WindowsAgent
Version: 6.0.0 (Build: 6.0.0-20111116-1426-b43)
Adapter ID: FglAM
Status: Active/Collecting data
Health State: OK
-----------------------------------
Host: Host2.example.com
ID: 5
Name: Monitor@Host2.example.com
Type: WindowsAgent
Version: 6.0.0 (Build: 6.0.0-20111116-1426-b43)
Adapter ID: FglAM
Status: Active/Not collecting data
Health State: OK
-----------------------------------
```

Activating an agent instance

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:activate -
agentid 9
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

deactivate

The `deactivate` command disables one or more Foglight agent instances.

If you want to delete an agent instance, you first stop the agent's data collection, deactivate it, and then delete the instance.

To delete an agent instance, a typical flow of actions requires you to first stop the agent's data collection, deactivate it, and then delete the instance. However, the `deactivate` command includes options and arguments, such as `-datacollection true`, that allow you to override that flow and agent instances that are collecting data.

For information on how to activate an agent instance using the *fglcmd* interface, see activate on page 63.

Scope

agent

Syntax

```
fglcmd connection_options -cmd agent:deactivate {-agentid agent_ID|-name
    agent_name|-namespace agent_namespace|-all} [-host host_name]
    [-adapter adapter_ID] [-type agent_type] [-deletable {true|false}]
    [-datacollection {true|false}] [-activatable {true|false}]
    [-active {true|false}] [-regex] [-force] [-nowait]
```

Table 15. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to deactivate agent instances that can (**true**) or cannot be deactivated (**false**).
active	**true** or **false**	Indicates whether to deactivate agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be deactivated.
all	None	Indicates that all agent instances should be deactivated.

Table 15. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to deactivate Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to deactivate Foglight agent instances that can (**true**) or cannot be deleted (**false**).
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be deactivated.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more Foglight agents that are to be deactivated.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
nowait	None	Indicates that the command should not wait for the selected agents to finish processing before deactivating the selected Foglight agent instances.
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Managing the Infrastructure Cartridge User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:deactivate -
agentid 14
```

i | **NOTE:** If successful, this command does not generate any output.

See also

- deploy on page 57
- create on page 60
- delete on page 61

Starting or Stopping Data Collection

start

The `start` command initiates data collection for one or more Foglight agent instances.

To create an agent instance that collects data, you first create the agent instance, activate it, and then start the data collection for that agent instance. Similarly, when you want to remove an agent instance, use a reverse order of actions: first stop the agent instance, deactivate it, and then delete it.

You should issue this command after creating and/or activating an agent instance. For information on how to activate agent instances using the `fglcmd` interface, see activate on page 63.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:start {-agentid agent_ID|-name agent_name|
    -namespace agent_namespace|-all} [-host host_name] [-adapter adapter_ID]
    [-type agent_type] [-deletable {true|false}] [-datacollection {true|false}]
    [-activatable {true|false}] [-active {true|false}] [-regex] [-force]
    [-nowait]
```

Table 16. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to start agent instances that can (**true**) or cannot be activated (**false**).
active	**true** or **false**	Indicates whether to start agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be started.
all	None	Indicates that all agent instances should be started.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to start Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to start Foglight agent instances that can (**true**) or cannot be deleted (**false**).

Table 16. Options and arguments

Option	Argument	Description
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be started.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more agent instances that are to be started.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
nowait	None	Indicates that the command should not wait for the selected agents to finish processing before starting the selected Foglight agent instances.
regex	None	Interprets the *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Managing the Infrastructure Cartridge User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:start -host
Host1 -active true -force
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

stop

The `stop` command terminates the data collection for one or more Foglight agent instances.

If you want to delete an agent instance, you first stop the agent's data collection, deactivate it, and then delete the instance.

To delete an agent instance, a typical flow of actions requires you to first stop the agent's data collection, deactivate it, and then delete the instance. However, the deactivate (see page 66) and delete (see page 61) commands that allow you to deactivate and delete agent instances, include options and arguments that let you override that flow and delete active agent instances that are collecting data without first stopping their data collection.

For information on how to start an agent's data collection using the command line, see start on page 68.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:stop {-agentid agent_ID|-name agent_name|
   -namespace agent_namespace|-all} [-host host_name] [-adapter adapter_ID]
   [-type agent_type] [-deletable {true|false}] [-datacollection {true|false}]
   [-activatable {true|false}] [-active {true|false}] [-regex] [-force]
   [-nowait]
```

Table 17. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to stop agent instances that can (**true**) or cannot be activated (**false**).
active	**true** or **false**	Indicates whether to stop agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent that is to be stopped.
all	None	Indicates that all agent instances should be stopped.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to stop Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to stop Foglight agent instances that can (**true**) or cannot be deleted (**false**).
force	None	In cases where multiple agents are selected, it indicates that all selected agents should be stopped.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies the name of one or more agent instances that are to be stopped.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
nowait	None	Indicates that the command should not wait for the selected agents to finish processing before stopping the selected Foglight agent instances.

Table 17. Options and arguments

Option	Argument	Description
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Foglight for Infrastructure User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:stop -agentid 15
```

i | **NOTE:** If successful, this command does not generate any output.

See also

- deploy on page 57
- create on page 60
- delete on page 61
- activate on page 63
- deactivate on page 66
- start on page 68

Listing and Retrieving Log Files

logs

The `logs` command shows a list of agent log files.

For information on how to retrieve a log file using the `fglcmd` interface, see getlog on page 73.

Scope

agent

Syntax

```
fglcmd connection_options -cmd agent:logs {-clientname display_name|-clientid
    client_ID|-clientbuild build_ID|-clientversion version|-allclients}
    [-host host_name] [-regex] [-force] [-upgradable {true|false}]
```

Table 18. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more instances of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
force	None	In cases where multiple client instances are selected, it indicates that all selected instances should be stopped.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
regex	None	Interprets the *build_ID, client_ID, display_name, host_name, version,* or *upgradable* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:logs -clientid
Host1#be05acc0-6179-4878-a274-afcd5075260a
Client ID: Host1#be05acc0-6179-4878-a274-afcd5075260a
Client Name: Host1
Host Name: Host1
   1 log files found.
   C:\Quest\Foglight\fglam\state\default\logs\HostAgents\6.0.0\
      WindowsAgent\Monitor@Host1_2011-11-21_151519_001.log
```

See also

- getlog on page 73

getlog

The `getlog` command retrieves a copy of an agent's log file. This command is useful in situations when you do not have access to the installation directory of the Foglight Management Server.

For information on how to get a list of log files using the `fglcmd` interface, see logs on page 71.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:getlog -log log_file_name -f file_path
   [-host host_name] [-clientname display_name] [-clientid client_ID]
   [-clientbuild build_ID] [-clientversion version] [-allclients] [-regex]
   [-upgradable {true|false}]
```

Table 19. Options and arguments

Option	Argument	Description
allclients	None	Indicates that all instances of the Foglight Agent Manager should be selected.
clientbuild	*build_ID*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified build ID.
clientid	*client_ID*	An explicit text string or a regular expression that identifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientname	*display_name*	An explicit text string or a regular expression that specifies one or more display devices that are running an instance of the Foglight Agent Manager.
clientversion	*version*	An explicit text string or a regular expression that selects one or more Foglight Agent Manager instances with the specified version.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the destination file.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
log	*log_file_name*	Specifies the path and name of the log file that is to be retrieved.
regex	None	Interprets the *build_ID, client_ID*, *display_name*, *host_name, version,* or *upgradable* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50.
upgradable	**true** or **false**	Indicates whether one or more Foglight Agent Manager instances can (**true**) or cannot be upgraded (**false**). It can be set to a regular expression.

Example

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd agent:getlog -clientid Host1#be05acc0-6179-4878-a274-afcd5075260a -log**

```
..fglam\state\default\logs\HostAgents\6.0.0\WindowsAgent\Monitor@Host1_2011- 11-
21_151519_001.log -f AgentA.log
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- logs on page 71

Listing and Assigning Blackout Schedules to Agent Instances

This section describes the following commands:

- showschedule on page 74
- setschedule on page 76

These commands allow you to assign blackout periods to agent instances, and to list them, as required. An agent blackout is a scheduled event during which the agent does not collect data for set intervals. Blackout periods can also be assigned to topology objects. However, unlike agent blackouts, topology object blackouts do not interrupt the data collection for the object to which the blackout is assigned. Blacking out a topology object simply means that no rules analyze that object for the duration of the blackout.

For more information about the commands for assigning blackouts to agent instances, and listing them, as required, see Listing and Assigning Blackout Schedules to Topology Objects on page 104.

showschedule

The `showschedule` command shows the blackout schedule assigned to one or more Foglight agent instances.

> **i** | **TIP:** Blackout periods prevent the data collection for the agent instance to which the blackout is assigned. Unlike agent blackouts, topology object blackouts prevent any rules from analyzing that object, without interrupting their data collection. For more information about topology object blackouts, see Listing and Assigning Blackout Schedules to Agent Instances on page 74.

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:showschedule {-agentid agent_ID|-name
    agent_name|-namespace agent_namespace|-all} [-host host_name]
    [-adapter adapter_ID] [-type agent_type] [-deletable {true|false}]
    [-datacollection {true|false}] [-activatable {true|false}]
    [-active {true|false}] [-regex]
```

Table 20. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to select agent instances that can (**true**) or cannot be deactivated (**false**).
active	**true** or **false**	Indicates whether to select agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent whose blackout schedule is to be set.
all	None	Indicates that all agent instances should be selected.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to select Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to select Foglight agent instances that can (**true**) or cannot be deleted (**false**).
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies one or more Foglight agents whose blackout schedule is to be set.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50. **NOTE:** On Unix systems, enclose the regular expression in quotation marks. Failing to do so can produce unpredictable results. On Windows, the quotation marks are optional.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Foglight for Infrastructure User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:showschedule -
active true
ID: 1
Host: Host1
```

```
Name: Monitor@Host1
Type: WindowsAgent
Status: Active/Collecting data
Blackout Schedules:
[none]
-----------------------------------
ID: 5
Host: Host2.example.com
Name: Monitor@Host2.example.com
Type: WindowsAgent
Status: Active/Not collecting data
Blackout Schedules:
        Nov 21, 2011 5:17:00 PM - Nov 22, 2011 6:17:00 PM (id: 05915e12-07a5-452
a-9d14-27b84bf45960) [in effect at the moment]
        Nov 21, 2011 5:17:00 PM - Nov 22, 2011 6:17:00 PM (id: 3734e17a-7201-429
c-9468-1af63b413e53) [in effect at the moment]
-----------------------------------
```

See also

- setschedule on page 76

setschedule

The `setschedule` command assigns a blackout schedule to one or more Foglight agent instances.

> **i** | **TIP:** Blackout periods prevent the data collection for the agent instance to which the blackout is assigned. Unlike agent blackouts, topology object blackouts prevent any rules from analyzing that object, without interrupting their data collection. For more information about topology object blackouts, see Listing and Assigning Blackout Schedules to Agent Instances on page 74.

For information on how to get a list of blackout schedules assigned to agent instances files using the `fglcmd` interface, see showschedule on page 74.

> **!** | **CAUTION: In addition to the features provided by the agent:setschedule fglcmd command, agent blackouts can also be configured using the Blackout Configuration dashboard in the browser interface. However, the mechanism for creating blackouts using this other method is independent. It is not recommended to use both methods on the same Foglight Management Server. If you choose to use the command line for creating blackouts, delete all blackouts created with the Blackout Configuration dashboard before using the command line. If you want to switch from the command line to the Blackout Configuration dashboard, use the conversion script to convert the existing blackouts created with the command line. This way all blackouts can be managed in one location. For more information about the conversion script, see the *Foglight Upgrade Guide*.**

Scope

`agent`

Syntax

```
fglcmd connection_options -cmd agent:setschedule {-agentid agent_ID|-name
    agent_name|-namespace agent_namespace|-all} {-schedulename schedule_name|
    -scheduleid schedule_ID -none [-remove]} [-host host_name]
    [-adapter adapter_ID] [-type agent_type] [-deletable {true|false}]
    [-datacollection {true|false}] [-activatable {true|false}]
    [-active {true|false}] [-regex] [-force]
```

Table 21. Options and arguments

Option	Argument	Description
activatable	**true** or **false**	Indicates whether to select agent instances that can (**true**) or cannot be deactivated (**false**).
active	**true** or **false**	Indicates whether to select agent instances that are active (**true**) or inactive (**false**).
adapter	*adapter_ID*	An explicit text string or a regular expression that identifies one or more Foglight adapters.
agentid	*agent_ID*	Identifies an instance of a Foglight agent whose blackout schedule is to be set.
all	None	Indicates that all agent instances should be selected.
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
datacollection	**true** or **false**	Indicates whether to select Foglight agent instances that are (**true**) or are not collecting data (**false**).
deletable	**true** or **false**	Indicates whether to select Foglight agent instances that can (**true**) or cannot be deleted (**false**).
force	None	In cases where multiple agents are selected, it indicates that the command should be performed against all selected agent instances.
host	*host_name*	An explicit text string or a regular expression that specifies one or more host names.
name	*agent_name*	An explicit text string or a regular expression that specifies one or more Foglight agents whose blackout schedule is to be set.
namespace	*agent_namespace*	An explicit text string or a regular expression that specifies the name of one or more Foglight agent instances in the specified name space.
none	None	Dissociates all schedules from one more selected agent instances.
regex	None	Interprets the *adapter_ID*, *agent_name*, *agent_namespace*, *agent_type*, or *host_name* arguments as regular expressions. For a sample regular expression used to specify a host name, see Example on page 50. **NOTE:** On Unix systems, enclose the regular expression in quotation marks. Failing to do so can produce unpredictable results. On Windows, the quotation marks are optional.
remove	None	Indicates that the specified schedule should be removed from one or more selected agent instances.
scheduleid	*schedule_ID*	Identifies a schedule that is to be assigned to one or more Foglight agent instances.

Table 21. Options and arguments

Option	Argument	Description
schedulename	*schedule_name*	Specifies a schedule name. You can use any of the following values: • *Daily Off Values* • *End of Day* • *Hourly* • *Monthly Off Hours* • *Start of Day* • *Quarterly Off Hours* • *Weekly Off Hours* **NOTE:** If you choose a schedule name that contains spaces, such as *Start Of Day,* enclose it in quotation marks.
type	*agent_type*	An explicit text string or a regular expression that specifies one or more Foglight agent types. The selection of available agent types depends on the collection of deployed cartridges. For example, a basic Foglight Management Server installation that includes the Cartridge for Operating Systems can offer a number of different agent types. For example, a basic Foglight Management Server installation that includes the Cartridge for Infrastructure can two different agent types: • WindowsAgent • UnixAgent For more information about these agents, see the *Foglight for Infrastructure User and Reference Guide*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd agent:setschedule -
agentid 10 -schedulename "Start of Day"
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

Installing or Uninstalling Cartridges

install

The `install` command installs a cartridge on the Foglight Management Server.

Scope

cartridge

Syntax

```
fglcmd connection_options -cmd cartridge:install [-passive] [-f file_path]
```

Table 22. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight™ Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
passive	None	Indicates that the cartridge should be installed, but not activated.
f	file_path	Specifies the path and file name of the cartridge file that is to be installed.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd cartridge:install -f
C:\Quest\carts\Infrastructure-5_8_0.car
Cartridges installed:
----
Name: Infrastructure version: 6.0.0 (enabled)
Name: HostAgents version: 6.0.0 (enabled)
```

See also

- uninstall on page 79
- disable on page 80
- enable on page 81
- list on page 82

uninstall

The uninstall command uninstalls a Foglight cartridge.

You can only uninstall a disabled cartridge. For information on how to disable a cartridge using the fglcmd interface, see disable on page 80.

Scope

cartridge

Syntax

```
fglcmd connection_options -cmd cartridge:uninstall [-n name] [-v version]
```

Table 23. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
n	*name*	Specifies the name of the cartridge that is to be uninstalled.
v	*version*	Specifies the version of the cartridge that is to be uninstalled.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd cartridge:uninstall -n
Infrastructure -v 6.0.0
```

i | **NOTE:** If successful, this command does not generate any output.

See also

- install on page 78
- disable on page 80
- enable on page 81
- list on page 82

Enabling or Disabling Cartridges

disable

The `disable` command deactivates a Foglight cartridge.

When you install a cartridge on the Foglight Management Server using default options either through the `fglcmd` interface or the Administration dashboards, Foglight activates that cartridge upon creation. For information on how to install a cartridge using the `fglcmd` interface, see install on page 78; to find out how to activate an inactive cartridge using `fglcmd`, see enable on page 81.

Scope

`cartridge`

Syntax

```
fglcmd connection_options -cmd cartridge:disable [-n name] [-v version]
```

Table 24. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
n	*name*	Specifies the name of the cartridge that is to be disabled.
v	*version*	Specifies the version of the cartridge that is to be disabled.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd cartridge:disable -n
Infrastructure -v 6.0.0
Cartridges disabled:
Infrastructure-6.0.0
```

See also

- install on page 78
- uninstall on page 79
- enable on page 81
- list on page 82

enable

The `enable` command activates an installed Foglight cartridge.

When you install a cartridge on the Foglight Management Server using default options either through the `fglcmd` interface or the Administration dashboards, Foglight activates that cartridge upon creation. For information on how to install a cartridge using the `fglcmd` interface, see install on page 78; to find out how to deactivate an active cartridge using `fglcmd`, see enable on page 81.

Scope

cartridge

Syntax

fglcmd *connection_options* **-cmd cartridge:enable** [**-n** *name*] [**-v** *version*]

Table 25. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
n	*name*	Specifies the name of the cartridge that is to be enabled.
v	*version*	Specifies the version of the cartridge that is to be enabled.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd cartridge:enable -n
Infrastructure -v 6.0.0
Cartridges enabled:
Infrastructure-6.0.0
```

See also

- install on page 78
- uninstall on page 79
- disable on page 80
- list on page 82

Listing Cartridges

list

The `list` command generates a list of all installed Foglight cartridges.

For information on how to install a Foglight cartridge using the `fglcmd` interface, see install on page 78.

Scope

cartridge

Syntax

fglcmd *connection_options* **-cmd cartridge:list**

Table 26. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd cartridge:list

Name: Core-ServiceLevelPolicy
Version: 6.0.0
Build: 6.0.0-5/13/2111161927-173845
Status: ACTIVATED
-----------------------------------
Name: Diagnostic
Version: 6.0.0
Build: 6.0.0-5/13/2111171301-147
Status: ACTIVATED
-----------------------------------
Name: FglAM-Adapter
Version: 5.6.2.2
Build: 5622-5/13/211116-1616-b30
Status: ACTIVATED
-----------------------------------
Name: Core-MonitoringPolicy
Version: 6.0.0
Build: 6.0.0-5/13/2111161927-173845
Status: ACTIVATED
-----------------------------------
…
```

See also

- install on page 78
- uninstall on page 79
- disable on page 80
- enable on page 81

Managing Security Entities

Foglight manages users' access through groups and roles. Each user belongs to one or more groups and each group can have one or more roles. The set of tasks that a user has access to depends on the roles that are assigned the groups that user belongs to.

This section contains reference information on commands that can be used to manage security settings in Foglight. For complete information about the security principles in Foglight, see the *Administration and Configuration Help*.

assigngroup

The `assigngroup` command adds or removes a Foglight user account from a group. A user can belong to one or more groups.

> **i** | **NOTE:** You can run this command only if the user account you are using to log into fglcmd has a Security role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the Administration and Configuration Help.

Scope

security

Syntax

fglcmd *connection_options* **-cmd security:assigngroup -username** *user_name* **-groupname** *group_name* [**-remove**]

Table 27. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
groupname	*group_name*	Specifies the name of the group to which the user account is to be added or removed.
remove	None	Indicates that the user should be removed from the group.
username	user_name	Specifies the name of the user account that is to be added or removed from the group.

Example

Adding a user to a group

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd security:assigngroup -username Demo -groupname MyGroup**

> **i** | **NOTE:** If successful, this command does not generate any output.

Removing a user from a group

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd security:assigngroup -username Demo -groupname MyGroup -remove**

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- createuser on page 86

- deleteuser on page 88

assignrole

The `assignrole` command adds or removes a Foglight role from a group. A group can have one or more roles.

> **i** | **NOTE:** You can run this command only if the user account you are using to log into fglcmd has a Security role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *connection_options* `-cmd security:assignrole -groupname` *group_name* `-rolename` *role_name* [`-remove`]

Table 28. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
groupname	*group_name*	Specifies the name of the group to which the role is to be added or removed.
remove	None	Indicates that the role should be removed from the group.
rolename	role_name	Specifies the name of the role that is to be added or removed from the group.

Example

Adding a role to a group

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:assignrole -groupname MyGroup -rolename Operator`**

> **i** | **NOTE:** If successful, this command does not generate any output.

Removing a role from a group

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:assignrole -groupname MyGroup -rolename Operator -remove`**

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- creategroup on page 86
- deletegroup on page 87

creategroup

The `creategroup` command creates a Foglight group. A group can have one or more users, and can be assigned one or more Foglight roles. When you create a group, that group has no roles assigned to it. Use the assignrole command to add a role to a group (see page 85).

> **i** | **NOTE:** You can run this command only if the user account you are using to log into `fglcmd` is granted the `Security` role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *connection_options* `-cmd security:creategroup -groupname` *group_name*

Table 29. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
groupname	*group_name*	Specifies the name of the group that is to be created.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd security:creategroup -groupname MyGroup
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- assignrole on page 85
- deletegroup on page 87

createuser

The `createuser` command creates a Foglight user account. When you create a user account, you can add it to a group using the **assigngroup** command (see page 84).

> **i** | **NOTE:** You can run this command only if the user account you are using to log into fglcmd has a Security role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *connection_options* `-cmd security:createuser -username` *user_name* `-password` *user_password*

Table 30. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
password	*user_password*	Specifies the password of the user account that is to be created.
username	*user_name*	Specifies the name of the user account that is to be created.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:createuser -username Demo -password Demo123`**

ℹ **NOTE:** If successful, this command does not generate any output.

See also

- assigngroup on page 84
- deleteuser on page 88

deletegroup

The `deletegroup` command removes a Foglight group.

ℹ **NOTE:** You can run this command only if the user account you are using to log into fglcmd has a Security role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *connection_options* `-cmd security:deletegroup -groupname` *group_name*

Table 31. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
groupname	*group_name*	Specifies the name of the group that is to be deleted.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:deletegroup -groupname MyGroup`**

> ℹ **NOTE:** If successful, this command does not generate any output.

See also

- assignrole on page 85
- creategroup on page 86

deleteuser

The `deleteuser` command removes a Foglight user account.

> ℹ **NOTE:** You can run this command only if the user account you are using to log into fglcmd has a Security role. For information about fglcmd connection options, see Logging In and Setting the Scope on page 46. For complete information about users, roles, and groups in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *`connection_options`* **`-cmd security:deleteuser -username`** *`user_name`*

Table 32. Options and arguments

Option	Argument	Description
username	*user_name*	Specifies the user name of the account that is to be deleted.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:deleteuser -username Demo`**

> ℹ **NOTE:** If successful, this command does not generate any output.

See also

-
-

exportldapcfg

The `exportldapcfg` command exports the existing Lightweight Directory Access Protocol (LDAP) settings to an XML file. LDAP settings enable access to users whose account information is stored in an external system. For more information about the LDAP settings in Foglight, see the *Administration and Configuration Help*. Once exported, the LDAP settings can be imported into Foglight using the importldapcfg command (see page 90).

> **i** | **TIP:** LDAP settings can be configured using the Configure Directory Services dashboard of the Administration module in the browser interface. For more information, see the *Administration and Configuration Help*.

Scope

security

Syntax

fglcmd *connection_options* **-cmd security:exportldapcfg** <**-f** *file_path*>

Table 33. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the XML file to which the LDAP settings are to be exported.

Example

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd security:exportldapcfg -f ldap.xml**

> **i** | **NOTE:** If successful, this command does not generate any output.

The content of the exported XML file is similar to the following content:

```
<?xml version="1.0" encoding="UTF-8"?>
<foglight-config version="1.0.0" exported-on="5/13/21-03-
   10T16:55:15.928Z">
   <sec-ldap-config>
      <id>3544adcb-9b5a-4f81-a347-8536d05c71ad</id>
      <ldap-svr-url>ldap://ldapsvr.example.com:389/</ldap-svr-url>
      <principal-dn-prefix>CN=</principal-dn-prefix>
      <principal-dn-suffix>,OU=Employees,DC=example,DC=com</
         principal-dn-suffix>
      <role-attr-id>name</role-attr-id>
```

```
    <role-attr-is-dn>false</role-attr-is-dn>
    <sec-svc-name>com.quest.nitro:service=NitroSecurityService
       </sec-svc-name>
    <ualias-attr-id>sAMAccountName</ualias-attr-id>
    <roles-ctx-dn>OU=Groups,DC=example,DC=com</roles-ctx-dn>
    <uid-attr-id>member</uid-attr-id>
    <match-on-user-dn>true</match-on-user-dn>
    <ldap-init-factory>com.sun.jndi.ldap.LdapCtxFactory</ldap-init-factory>
    <ldap-auth-type>simple</ldap-auth-type>
    <login-module-name>com.quest.nitro.service.security.auth.
       spi.NitroExtendedLdapLoginModule</login-module-name>
    <sec-domain-name>fgl-web-console</sec-domain-name>
    <parent-group-attr-id>memberOf</parent-group-attr-id>
    <gid-attr-id>member</gid-attr-id>
    <search-order>0</search-order>
    <bind-user-dn/>
    <bind-credential/>
    <user-ctx-dn>ou=Employees,DC=example,DC=com</user-ctx-dn>
    <ldap-svr-url2>ldap://backupldapsvr.example.com:389/</ldap-svr-url2>
    <role-ctx-dn2>OU=Dynamic Groups,DC=example,DC=com</role-ctx-dn2>
    <role-ctx-dn3/>
    <max-rec-depth>15</max-rec-depth>
    <search-time-limit>10000</search-time-limit>
    <role-search-mode>direct</role-search-mode>
    <bind-pwd-prefix>bindpwd.</bind-pwd-prefix>
  </sec-ldap-config>
</foglight-config>
```

See also

importldapcfg

The `importldapcfg` command imports LDAP settings to Foglight. The source of the import is an XML file that has been previously exported with the exportldapcfg command (see page 89). LDAP settings enable access to users whose account information is stored in an external system. For more information about the LDAP settings in Foglight, see the *Administration and Configuration Help*.

i | **TIP:** LDAP settings can be configured using the Configure Directory Services dashboard of the Administration module in the browser interface. For more information, see the *Administration and Configuration Help*.

Scope

security

Syntax

fglcmd *connection_options* **-cmd security:importldapcfg -f** *file_path>*

Table 34. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the LDAP settings file that is to be installed. Specifies the path and name of the XML file containing the LDAP settings that is to be imported.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd security:importldapcfg -f ldap.xml`**

ℹ **NOTE:** If successful, this command does not generate any output.

See also

- exportldapcfg on page 89

list

The `list` command lists users, groups, and/or roles, and indicates their relationships, as specified by a combination of the available options.

ℹ **TIP:** A user can belong to one or more groups while each group can have one or more roles. Belonging to a group provides the users with access to all of the roles that are assigned to that group. For complete information about the security principles in Foglight, see the *Administration and Configuration Help*.

Scope

`security`

Syntax

`fglcmd` *`connection_options`* **`-cmd security:list [-g] [-r] [-u] [-v]`**

Table 35. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
g	None	Lists all groups that exist in Foglight.
r	None	Lists all roles that exist in Foglight.

Table 35. Options and arguments

Option	Argument	Description
u	None	Lists all users that exist in Foglight.
v	None	For each listed entity (user, group, and/or role), this option shows its relationships with other entity types (user, group, and/or role). For example, when listing users using the **u** option, this option also list the groups and roles associated with each user.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd security:list -u -g -r
-v
Users:
   Name: foglight, External?: false, Status: A
      [G] Cartridge Developers
      [G] Foglight Security Administrators
      [G] Foglight Administrators
      [R] Console User
      [R] Cartridge Developer
      [R] General Access
      [R] Operator
      [R] Dashboard User
      [R] Advanced Operator
      [R] Security
      [R] Dashboard Designer
      [R] Administrator

Groups:
   Name: Foglight Administrators, External?: false
      [U] foglight
      [R] Dashboard Designer
      [R] Administrator
      [R] Operator
      [R] Dashboard User
      [R] General Access
      [R] Console User
      [R] Advanced Operator
   Name: Foglight Operators, External?: false
      [R] Operator
      [R] Console User
   Name: Foglight Security Administrators, External?: false
      [U] foglight
      [R] Security
   Name: Cartridge Developers, External?: false
      [U] foglight
      [R] Cartridge Developer
   [R] Console User

Roles:
   Name: Dashboard Designer, Type: T
      [G] Foglight Administrators
   Name: Security, Type: T
      [G] Foglight Security Administrators
   Name: Administrator, Type: T
      [G] Foglight Administrators
   Name: Cartridge Developer, Type: T
      [G] Cartridge Developers
   Name: Console User, Type: T
      [G] Foglight Operators
```

```
   [G] Cartridge Developers
   [G] Foglight Administrators
Name: Dashboard User, Type: T
   [G] Foglight Administrators
Name: General Access, Type: T
   [G] Foglight Administrators
Name: Advanced Operator, Type: T
   [G] Foglight Administrators
Name: Operator, Type: T
   [G] Foglight Operators
   [G] Foglight Administrators
```

Managing Foglight Licenses

Foglight allows you to access only those product features that are defined in your license file, and in some cases, during a specific time period. Your Foglight environment requires a license for the server and a license for each cartridge that is license-protected. Some Foglight cartridges are license-protected, while others do not require a license.

You can install any cartridges, regardless of whether they are listed in your Foglight license. Installing a license-protected cartridge without adding its license results in that cartridge being disabled by default. This section describes the `fglcmd` commands that you can use to manage Foglight licenses. For more information about the licensing capabilities, see the *Administration and Configuration Help*.

import

The `import` command installs a Foglight license. Use it to install a Foglight license file whose name and location are specified by the argument.

For information on how to remove a license using the `fglcmd` interface, see remove on page 94.

Scope

`license`

Syntax

fglcmd *connection_options* **-cmd license:import -f** *<file_path>*

Table 36. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and file name of the license file that is to be installed.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd license:import -f
..\license\foglight.license
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- remove on page 94
- list on page 94

list

The `list` command generates a list of installed Foglight licenses.

Scope

`license`

Syntax

fglcmd *connection_options* **-cmd license:list**

Table 37. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd license:list
License Serial: 123-4567890
License Expiration Date: Fri Aug 01 00:00:00 EDT 5/13/21
Licensed Server Features:
    agents_connection
    ldap_integration
    config_management
    performance_calendars
    request_trace_analysis
    cartridge_installation
    data_archiving
    high_availability

Agent Licenses:
----------------------------------
```

remove

The `remove` command deletes a Foglight license given a license serial number.

For information on how to add a Foglight license using the `fglcmd` interface, see import on page 93.

Scope

`license`

Syntax

`fglcmd` `connection_options` `-cmd license:remove` [`-serial` `serial_number`]

Table 38. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
serial	*serial_number*	Specifies the serial number of the license file that is to be removed.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd license:remove -serial`**
`123-4567890`

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- import on page 93
- list on page 94

Exporting or Importing a Monitoring Policy

configexport

The `configexport` command exports the monitoring policy to an XML file. A monitoring policy describes all customizations to rules, derived metrics, registry variables, agent settings and persistence policies. It is useful to export this information for technical analysis and modification purposes even though that data is not meant to be readable by end-users. The `configexport` command saves the current monitoring policy to an XML file given the file path and its name. If required, you can use the output file to re-import the monitoring policy at a later time using the configimport command (see page 96)

Scope

`util`

Syntax

`fglcmd` *connection_options* `-cmd util:configexport` [`-f` *file_path*]

Table 39. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and file name of the XML file to which the monitoring policy is to be exported. **NOTE:** When specifying the file name, use `XML` as the file extension.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd util:configexport -f policy.xml`**

> **NOTE:** If successful, this command does not generate any output.

See also

* configimport on page 96

configimport

The `configimport` command imports a monitoring policy. The source of the import is an XML file that has been previously exported with the configexport command (see page 95). A monitoring policy describes all customizations to rules, derived metrics, registry variables, agent settings and persistence policies and is useful in technical analysis and modification tasks.

Any rules that you import using this command are not associated with any cartridges on the target system after the import. For example, if you use this command to move cartridge-deployed rules from one Management Server to another, even though the target location may contain the same collection of cartridges, the imported rules are not associated with any of these cartridges in the target location. Only those rules that are deployed by a cartridge can be associated with the cartridge.

Scope

`util`

Syntax

`fglcmd` *connection_options* `-cmd util:configimport` [`-f` *file_path*]

Table 40. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and file name of the XML file that is to be imported.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:configimport -f
policy.xml
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- configexport on page 95

Exporting, Importing, or Listing UI Modules

uiexport

The `uiexport` command exports a UI module to a ZIP file. Use this command to export any modifications to Foglight operational elements such as views, dashboards, and other resources. Given a valid module name and a file path and name, the `uiexport` command exports the contents of that module into a ZIP file. If required, you can use the output file to re-import that module into the same or a different Foglight server at a later time using the uiimport command (see page 98).

Scope

`util`

Syntax

`fglcmd` *connection_options* `-cmd util:uiexport` [`-m` *module_name*] [`-f` *file_path*]

Table 41. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the ZIP file that the UI module is to be exported to. **NOTE:** When specifying the file name, use `ZIP` as the file extension.
m	*module_name*	Specifies the name of the UI module.

Example

The following example shows you how to export a module to a ZIP file using the module name as an argument. To get a list of module names, use the uilist command (see page 99), as shown in Example on page 99.

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight-cmd util:uiexport -m
system:fsmagents -f c:\temp\sys_fsmagents.zip
```

> **NOTE:** If successful, this command does not generate any output.

See also

- uiimport on page 98
- uilist on page 99

uiimport

The `uiimport` command imports a UI module from a ZIP file. It uses the ZIP file that has been previously imported with the uiexport command (see page 97) as the source of import. A UI module is a collection of Foglight operational elements such as views, dashboards, and other resources.

Scope

`util`

Syntax

```
fglcmd connection_options -cmd util:uiimport [-f file_path]
```

Table 42. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the ZIP file containing the UI module that is to be imported.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:uiimport -f
c:\temp\sys_fsmagents.zip
```

> **i** | **NOTE:** If successful, this command does not generate any output.

See also

- uiexport on page 97
- uilist on page 99

uilist

The `uilist` command shows a list of deployed UI modules. Each UI module is a collection of Foglight operational elements such as views, dashboards, and other resources.

Scope

util

Syntax

fglcmd *connection_options* **-cmd util:uilist**

Table 43. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:uilist
system:schemadatasource_typelist
system:administration_blackout_configuration_edit20properties
```

```
system:administration_rulesnotifications_rulemanagement
system:administration_grouping_retention
system:foglight_services_perspectives
…
```

See also

- uiexport on page 97
- uiimport on page 98

Exporting Metrics and Topology

metricexport

The `metricexport` command exports metric observations to a file using a metric query. String and metric observations that exist in Foglight can be retrieved with this command for automation and analysis. This command writes its output using a CSV or XML format and takes an observation query as a parameter.

> **i** | **NOTE:** Query writing requires an understanding of the monitored topology and object naming conventions and depends on data availability for the specified collection period (see Example on page 101). For complete information on topology queries, see the *Administration and Configuration Help*.

Scope

`util`

Syntax

fglcmd *connection_options* **-cmd util:metricexport -output_format {xml|csv} -metric_query** *query* **-f** *file_path*

Table 44. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the file into which the results of the query are to be exported. **NOTE:** When specifying the file name, use `XML` or `CSV` as the file extension, depending on the setting of the output_format option.
metric_query	*query*	Contains the metric query. Typically, a metric query specifies topology and object names and collection periods as parameters to retrieve metric observations. For full syntax information, see the *Administration and Configuration Help*.
output_format	**xml** or **csv**	Defines the output format of the file the query is to be exported into: XML or CSV.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:metricexport -
output_format csv -metric_query "Utilization from System_Table for 1 hour" -f
my_metric_query.csv
```

> **i** | **NOTE:** If successful, this command does not generate any output in the Command Prompt window (Windows) or the terminal window (Unix or Linux).

The above command creates a CSV file showing the value of the `Utilization` metric for all known `System_Table` objects that were collected in the past hour. The content of the generated CSV file are similar to the information in the following table:

Table 45. CSV file contents

uniqueId	startTime	endTime	samplePeriod	count	min	max	avg	sum	sumSquares	stdDev
a307df8f-63e9-4b65-a6d6-a50524a5544b	21:58.0	26:58.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	26:59.0	31:59.0	300000	1	2	2	2	2	4	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	32:00.0	37:00.0	300000	1	2	2	2	2	4	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	37:00.0	42:00.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	42:01.0	47:01.0	300000	1	2	2	2	2	4	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	47:02.0	52:02.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	52:02.0	57:02.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	57:03.0	02:03.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	02:03.0	07:03.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	07:04.0	12:04.0	300000	1	4	4	4	4	16	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	12:05.0	17:05.0	300000	1	1	1	1	1	1	0
a307df8f-63e9-4b65-a6d6-a50524a5544b	17:05.0	22:05.0	300000	1	8	8	8	8	64	0
ac205363-863b-478b-b6b7-97e1a444e76a	33:12.0	38:12.0	300000	1	1	1	1	1	1	0
ac205363-863b-478b-b6b7-97e1a444e76a	38:12.0	43:12.0	300000	1	1	1	1	1	1	0
ac205363-863b-478b-b6b7-97e1a444e76a	43:13.0	48:13.0	300000	1	2	2	2	2	4	0
ac205363-863b-478b-b6b7-97e1a444e76a	48:13.0	53:13.0	300000	1	1	1	1	1	1	0
ac205363-863b-478b-b6b7-97e1a444e76a	53:14.0	58:14.0	300000	1	2	2	2	2	4	0
ac205363-863b-478b-b6b7-97e1a444e76a	58:15.0	03:15.0	300000	1	1	1	1	1	1	0
ac205363-863b-478b-b6b7-97e1a444e76a	03:15.0	08:15.0	300000	1	1	1	1	1	1	0
ac205363-863b-478b-b6b7-97e1a444e76a	08:16.0	13:16.0	300000	1	4	4	4	4	16	0
ac205363-863b-478b-b6b7-97e1a444e76a	13:16.0	18:16.0	300000	1	3	3	3	3	9	0
ac205363-863b-478b-b6b7-97e1a444e76a	18:17.0	23:17.0	300000	1	6	6	6	6	36	0

topologyexport

The `topologyexport` command exports the value of one or more properties of a topology object to an XML file. Given a topology query, the property name, and the path and name of the output file, this command generates an XML file containing the value of the specified property. For complete information on how to write a topology query, see the *Administration and Configuration Help*.

Scope

`util`

Syntax

```
fglcmd connection_options -cmd util:topologyexport -property_names properties
-topology_query "query" -f file_path
```

Table 46. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the file into which the results of the query are to be exported. **NOTE:** When specifying the file name, use `XML` as the file extension.
property_names	*properties*	Contains one or more property names, separated by commas.
topology_query	*"query"*	Contains the topology query. Typically, a topology query specifies topology and object names as parameters to retrieve one or more object instances. The query argument must be enclosed in double quotes. Any variables in the query must be enclosed in single quotes. For example: `"Windows_Host where name='host1'"` To learn more about the query syntax , see the *Administration and Configuration Help*.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:topologyexport -f
host1_alarm_count.xml -topology_query "Windows_Host where name='host1'" -
property_names alarmTotalCount
```

> **i** | **NOTE:** If successful, this command does not generate any output in the Command Prompt window (Windows) or the terminal window (Unix or Linux).

The above command generates an XML file showing the value of the `alarmTotalCount` property for a particular Windows[®] host. The content of the generated XML file is similar to the following listing:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<top-objects>
<top-obj uniqueId="c380ae79-2d36-45fc-a07f-136509fec207">
<property name="uniqueId" value="c380ae79-2d36-45fc-a07f-136509fec207"/>
<property name="topologyObjectId" value="175"/>
<property name="topologyObjectVersionId" value="280"/>
<property name="topologyObjectVersion" value="1"/>
<property name="effectiveStartDate" value="5/13/21-01-02 17:29:14.82"/>
<property name="effectiveEndDate" value="5138-11-16 04:46:40.0"/>
<property name="lastUpdated" value="5/13/21-01-02 17:29:14.929"/>
<property name="name" value="host1.example.com"/>
<property name="longName" value="host1.example.com (Windows_Host)"/>
<property name="scheduleIds" value="[]"/>
```

```
<property name="isBlackedOut" value="false"/>
<property name="annotations" value="[]"/>
<property name="alarms" value="[]"/>
<property name="aggregateAlarms" value="[]"/>
<property name="localState" value="0"/>
<property name="aggregateState" value="0"/>
<property name="aggregateAlarmState" value="DataObject of type 'foglight-
5:AlarmStateObservation' for datasource: 'foglight-5:foglight-
5':/observations/c380ae79-2d36-45fc-a07f-136509fec207/aggregateAlarmState"/>
<property name="alarmWarningCount" value="0"/>
<property name="alarmCriticalCount" value="0"/>
<property name="alarmFatalCount" value="0"/>
<property name="alarmTotalCount" value="0"/>
<property name="alarmAggregateWarningCount" value="0"/>
<property name="alarmAggregateCriticalCount" value="0"/>
<property name="alarmAggregateFatalCount" value="0"/>
<property name="alarmAggregateTotalCount" value="0"/>
<property name="changeSummary" value="[]"/>
<property name="changeCount" value="0"/>
<property name="aggregateChangeCount" value="0"/>
<property name="topologyTypeName" value="Windows_Host"/>
<property name="monitoredHost" value="foglight-5:Windows_Host:c380ae79-2d36-45fc-
a07f-136509fec207:1 datasource=foglight-5:foglight-5"/>
<property name="sourceIds" value="[]"/>
<property name="serviceLevelPolicies" value="[]"/>
<property name="ipAddresses" value="[]"/>
<property name="interfaces" value="[]"/>
<property name="running" value="[]"/>
<property name="detail" value="[]"/>
<property name="agents" value="[]"/>
</top-obj>
</top-objects>
```

Looking Up Server Parameters

env

The `env` command shows the values of server configuration parameters. The parameters are specified in *<foglight_home>/config/server.config*. You can output the values of all configuration parameters, or use an option to specify a single parameter.

Scope

`util`

Syntax

fglcmd *connection_options* **-cmd util:env** [**-n** *parameter_name*]

Table 47. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
n	*parameter_name*	Specifies the name of the parameter whose value is to be retrieved. If you do not provide a parameter name, the command lists all of the configuration parameters.

Example

Displaying the Foglight database port number

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd util:env -n
foglight.database.port
13306
```

Listing and Assigning Blackout Schedules to Topology Objects

This section describes the following commands:

- blackoutobject on page 104

- blackouts on page 106

These commands allow you to assign blackout periods to topology object instances, and to list them, as required. A topology object blackout is a scheduled event during which no rules analyze that object for the duration of the blackout. Topology object blackouts do not interrupt the data collection for the object to which the blackout is assigned. Blackout periods can also be assigned to agent instances. However, unlike topology object blackouts, agent blackouts prevent their data collection.

For more information about the commands for assigning blackouts to agent instances, and listing them, as required, see Listing and Assigning Blackout Schedules to Agent Instances on page 74.

blackoutobject

The `blackoutobject` command assigns a blackout schedule to topology objects. A blackout schedule defines the periods of time during which no alarms are raised for a specified object instance. The command uses either the schedule name or its ID to assign one or more topology objects that are specified either by their IDs, or a topology query. Blacking out a topology object means that no rules analyze that object for the duration of the blackout.

> **i** | **TIP:** Blackout periods do not interrupt the data collection for the object to which the blackout is assigned. Unlike topology object blackouts, agent blackouts prevent their data collection. For more information about agent blackouts, see Listing and Assigning Blackout Schedules to Agent Instances on page 74.

This command can make use of topology queries to retrieve one or more object instances which allows you to automate blackouts (for example, cron-driven changes in blackout policies). For complete information on how to write a topology query, see the *Administration and Configuration Help*.

> **CAUTION:** **In addition to the features provided by the topology:blackoutobject fglcmd command, topology blackouts can also be configured using the Blackout Configuration dashboard in the browser interface. However, the mechanism for creating blackouts using this other method is independent. It is not recommended to use both methods on the same Foglight™ Management Server. If you choose to use the command line for creating blackouts, delete all blackouts created with the Blackout Configuration dashboard before using the command line. If you want to switch from the command line to the Blackout Configuration dashboard, use the conversion script to convert the existing blackouts created with the command line. This way all blackouts can be managed in one location. For more information about the conversion script, see the *Foglight Upgrade Guide*.**

Scope

topology

Syntax

fglcmd *connection_options* **-cmd topology:blackoutobject** {**-object** *object_ID* | **-query** "*topology_query*"} [**-remove**] {**-schedule** *schedule_ID* | **-schedulename** *schedule_name*} [**-inheritable**] [**-clear**]

Table 48. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, fglcmd can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
clear	None	Instructs Foglight to remove any schedules associated with one or more specified topology objects.
inheritable	None	Indicates that the blackout schedule you are about to assign to the selected topology object also applies to the descendents of those objects.
object	*object_ID*	Identifies the topology object.

Table 48. Options and arguments

Option	Argument	Description

query	*topology_query*	Contains the topology query that selects one or more topology objects. Typically, a topology query uses topology types and object names as parameters to retrieve one or more object instances. To find out more about the syntax for writing queries, see the *Administration and Configuration Help*.
		
		NOTE: The query must be enclosed in quotation marks. Failing to do so can produce unexpected results.
		
		Additionally, on Unix platforms, for query expressions that select objects of the `Host` type and start with an exclamation mark '!', the exclamation mark must be escaped with a back slash '\' for some Unix shells. For example:
		
		Unix and zsh
		`"\!Host where name like 'host1%'"`
		
		Unix and bash/sh
		`"!Host where name like 'host1%'"`
		
		Windows
		`"!Host where name like 'host1'"`

remove	None	Indicates that the specified schedule should be removed from one or more specified objects.
schedule	*schedule_ID*	Identifies the blackout schedule. To find out the ID for a schedule, use the list command. For more information, see list on page 108.
schedulename	*schedule_name*	Specifies the schedule name.

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd
topology:blackoutobject -query "Host where name ='host1.example.com'" -schedulename
Always
```
Updated the following topology objects:
d48afea1-7294-46e4-bcec-d486d20021be: Host: host1.example.com

> **ⓘ** | **TIP:** To find out the names and IDs of the available schedules, prior to issuing topology:blackoutobject, you can use the schedule:list command. For more information, see Listing Schedules on page 108.

See also

- blackouts on page 106
- list on page 108

blackouts

The `blackouts` command lists all blackouts currently assigned to topology objects. A blackout schedule defines the periods of time during which no alarms are raised for a specified object instance.

> **ⓘ** | **TIP:** Blackout periods do not interrupt the data collection for the object to which the blackout is assigned. Unlike topology object blackouts, agent blackouts prevent their data collection. For more information about agent blackouts, see Listing and Assigning Blackout Schedules to Agent Instances on page 74.

Blackout schedules can be assigned to object instances using the blackoutobject command (see page 104).

Scope

`topology`

Syntax

`fglcmd` *connection_options* `-cmd topology:blackouts` [`-query "`*topology_query*`"`]

Table 49. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
query	*topology_query*	Contains the topology query that selects one or more topology objects. Typically, a topology query uses topology types and object names as parameters to retrieve one or more object instances. To find out more about the syntax for writing queries, see the *Administration and Configuration Help*.
		NOTE: The query must be enclosed in quotation marks. Failing to do so can produce unexpected results.
		Additionally, on Unix platforms, for query expressions that select objects of the `Host` type and start with an exclamation mark '!', the exclamation mark must be escaped with a back slash '\'. For example:
		Unix `"\!Host where name like 'host1%'"`
		Windows `"!Host where name like 'host1'"`

Example

```
C:\Quest\Foglight\bin>fglcmd -usr foglight -pwd foglight -cmd topology:blackouts -
query "Host where name = 'host1.example.com'"
Blackouts for objects:
d48afea1-7294-46e4-bcec-d486d20021be: Host: host1.example.com, Always
```

> **TIP:** To find out the names and IDs of the available schedules, you can issue the schedule:list command. For more information, see Listing Schedules on page 108.

See also

- blackoutobject on page 104
- list on page 108

Listing Schedules

list

The `list` command generates a list of all Foglight schedules.

Scope

schedule

Syntax

fglcmd *connection_options* **-cmd schedule:list**

Table 50. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.

Example

C:\Quest\Foglight\bin>**fglcmd -usr foglight -pwd foglight -cmd schedule:list**

```
ID: 259833b6-d7b0-4b27-a5c4-e2e612cbf77a
Name: Frequent [Test]
Description: A schedule that runs very frequently. Primarily
used for testing.This schedule is used to trigger an activity.
As a result it does not have a significant duration.
------------------------------------
ID: 30afa315-3131-48c2-b0e8-71fea187af4e
Name: Always
Description: A schedule that includes all the time
------------------------------------
ID: 013f57d1-1e7e-4098-a7a5-2b2763364cb5
Name: First day of week
Description: Whole days Monday
------------------------------------
ID: 004c98d4-c278-4af2-81f7-4264fd30003f
Name: Daily Database Maintenance
Description: Schedule on which daily operations to rollup and
purge information in the database are performed. This schedule
is used to trigger an activity. As a result it does not have
a significant duration.
------------------------------------
ID: d7619930-807e-406e-9e6e-15ec30edc23c
Name: End of Day
Description: A schedule that runs at the end of the day. This
schedule is used to trigger an activity. As a result it does
```

```
not have a significant duration.
-------------------------------------
ID: 46fc602a-8931-4907-a93b-e1c16c0fdbb5
Name: Start of Day
Description: A schedule that runs at the start of the day.
This schedule is used to trigger an activity. As a result it
does not have a significant duration.
-------------------------------------
ID: eb7e213d-f14d-4d09-ba9d-2a266274d443
Name: Monthly Off Hours
Description: A schedule that runs every month off hours. This
schedule is used to trigger an activity. As a result it does
not have a significant duration.
-------------------------------------
ID: 0749990a-c878-4d1d-8002-fc80b9a31bd6
Name: Beginning of the day
Description: 00:00AM of every day
-------------------------------------
ID: deaf446f-bc2b-4ff5-b351-137baef609ff
Name: Quarterly Off Hours
Description: A schedule that runs at the start of every
quarter, off hours. This schedule is used to trigger an
activity. As a result it does not have a significant
duration.
-------------------------------------
ID: 3e3e1877-7b55-4c7a-9a40-975bdc96f8e2
Name: First day of month
Description: Whole days the first day of every month
-------------------------------------
ID: 1e698670-f5e1-449e-94ba-942a76795602
Name: Hourly
Description: A schedule that runs every hour. This schedule
is used to trigger an activity. As a result it does not have
a significant duration.
-------------------------------------
ID: a74dc804-4b42-4589-be5f-0aab6ca568d3
Name: Business hours
Description: 9AM to 5PM Monday to Friday
-------------------------------------
ID: d6bc50e7-0748-4356-909e-404aa7f377ef
Name: Daily Off Hours
Description: A schedule that runs every day off hours. This
schedule is used to trigger an activity. As a result it does
not have a significant duration.
-------------------------------------
ID: d601fef9-dd66-4584-af72-e4660138e93e
Name: Beginning of the month
Description: 00:00AM of the first day of every month
-------------------------------------
ID: ce5bcb30-c5d8-4388-89aa-8da82b8c666b
Name: Business week
Description: Whole days Monday to Friday
-------------------------------------
ID: e6816241-a745-4763-84ab-77766a2b5049
Name: Beginning of the week
Description: 00:00AM of every Monday
-------------------------------------
ID: 670bb364-7a3a-44ba-80b7-58419c060496
Name: Weekly Off Hours
Description: A schedule that runs every week off hours. This
schedule is used to trigger an activity. As a result it does
```

```
not have a significant duration.
-----------------------------------
ID: 94e5034b-42f1-448d-8b88-291915234109
Name: Weekends
Description: Whole days Saturday and Sunday
-----------------------------------
```

Running Scripts

run

The `run` command runs a script.

Typically, this command is used to run scripts that are deployed with Foglight cartridges, and is used in maintenance and support tasks when required.

Scope

`script`

Syntax

fglcmd *connection_options* **-cmd script:run -f** *file_path*

Table 51. Options and arguments

Option	Argument	Description
f	*file_path*	Specifies the path and name of the script file that is to be executed.

Example

`C:\Quest\Foglight\bin>`**fglcmd -usr foglight -pwd foglight -cmd script:run -f my.script**

> **i** | **NOTE:** If successful, this command does not generate any output in the Command Prompt window (Windows) or the terminal window (Unix or Linux).

Creating Server Support Bundles

bundle

The `bundle` command generates a server support bundle file in ZIP format.

Scope

`support`

Syntax

```
fglcmd connection_options -cmd support:bundle -f file_path
```

Table 52. Options and arguments

Option	Argument	Description
connection_options		Specifies the user name and password for the Foglight Management Server, as well as the machine name and port number, if necessary. Alternatively, `fglcmd` can retrieve this information from a user-specific properties file. For more information on how to specify connection options on the command line, see Logging In and Setting the Scope on page 46. For details on using a properties file, see Using a Properties File to Supply Connection Information on page 48.
f	*file_path*	Specifies the path and name of the support bundle file that is to be generated. **NOTE:** When specifying the file name, use `ZIP` as the file extension.

Example

`C:\Quest\Foglight\bin>`**`fglcmd -usr foglight -pwd foglight -cmd support:bundle -f support.bundle`**

ℹ | **NOTE:** If successful, this command does not generate any output.

About Us

Quest creates software solutions that make the benefits of new technology real in an increasingly complex IT landscape. From database and systems management, to Active Directory and Office 365 management, and cyber security resilience, Quest helps customers solve their next IT challenge now. Around the globe, more than 130,000 companies and 95% of the Fortune 500 count on Quest to deliver proactive management and monitoring for the next enterprise initiative, find the next solution for complex Microsoft challenges and stay ahead of the next threat. Quest Software. Where next meets now. For more information, visit https://www.quest.com/.

Technical support resources

Technical support is available to Quest customers with a valid maintenance contract and customers who have trial versions. You can access the Quest Support Portal at https://support.quest.com.

The Support Portal provides self-help tools you can use to solve problems quickly and independently, 24 hours a day, 365 days a year. The Support Portal enables you to:

- Submit and manage a Service Request.
- View Knowledge Base articles.
- Sign up for product notifications.
- Download software and technical documentation.
- View how-to-videos.
- Engage in community discussions.
- Chat with support engineers online.
- View services to assist you with your product.