
Micro Focus Security ArcSight SmartConnector

Action Connector Developer's Guide

Document Release Date: December 2021

Software Release Date: December 2021



Legal Notices

Micro Focus
The Lawn
22-30 Old Bath Road
Newbury, Berkshire RG14 1QN
UK

<https://www.microfocus.com>

Copyright Notice

© Copyright 2016-2021 Micro Focus or one of its affiliates

Confidential computer software. Valid license from Micro Focus required for possession, use or copying. The information contained herein is subject to change without notice.

The only warranties for Micro Focus products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Micro Focus shall not be liable for technical or editorial errors or omissions contained herein.

No portion of this product's documentation may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording, or information storage and retrieval systems, for any purpose other than the purchaser's internal use, without the express written permission of Micro Focus.

Notwithstanding anything to the contrary in your license agreement for Micro Focus ArcSight software, you may reverse engineer and modify certain open source components of the software in accordance with the license terms for those particular components. See below for the applicable terms.

U.S. Governmental Rights. For purposes of your license to Micro Focus ArcSight software, "commercial computer software" is defined at FAR 2.101. If acquired by or on behalf of a civilian agency, the U.S. Government acquires this commercial computer software and/or commercial computer software documentation and other technical data subject to the terms of the Agreement as specified in 48 C.F.R. 12.212 (Computer Software) and 12.211 (Technical Data) of the Federal Acquisition Regulation ("FAR") and its successors. If acquired by or on behalf of any agency within the Department of Defense ("DOD"), the U.S. Government acquires this commercial computer software and/or commercial computer software documentation subject to the terms of the Agreement as specified in 48 C.F.R. 227.7202-3 of the DOD FAR Supplement ("DFARS") and its successors. This U.S. Government Rights Section 18.11 is in lieu of, and supersedes, any other FAR, DFARS, or other clause or provision that addresses government rights in computer software or technical data.

Trademark Notices

Adobe™ is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number
- Document Release Date, which changes each time the document is updated
- Software Release Date, which indicates the release date of this version of the software

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

<https://www.microfocus.com/support-and-services/documentation>

Support

Contact Information

Phone	A list of phone numbers is available on the Technical Support Page: https://softwaresupport.softwaregrp.com/support-contact-information
Support Web Site	https://softwaresupport.softwaregrp.com/
ArcSight Product Documentation	https://community.softwaregrp.com/t5/ArcSight-Product-Documentation/ct-p/productdocs

Revision History

Date	Product Version	Description
11/30/2016	2.0	HPE ArcSight rebranding. Removed references to NSP/TRM as these products are no longer supported.
02/15/2012	1.0	First release of this guide.

Contents

- Chapter 1: Overview 7
 - Action Connector Development Options 7

- Chapter 2: Installing the Flex CounterACT Connector 9

- Chapter 3: Flex CounterACT Connector Properties File 11

- Chapter 4: Flex CounterACT Connector Examples 12
 - Nmap Example 12
 - Advanced Nmap Example 14

- Chapter 5: Integration Commands 17
 - What are Integration Commands? 17
 - Supported Command Types 18
 - Out-of-the-Box Commands for Logger 18
 - Local Scripts and Commands to Other Applications 18
 - How it Works 19
 - Planning Checklist and Workflow 19
 - Navigating to Integration Command Resources 21
 - Quick Example 21
 - Defining Commands 24
 - Command Types and Attributes 25
 - Script Commands 26
 - URL Commands 27
 - Connector Commands 28
 - Adding and Editing Command Parameters 29
 - Removing a Command Parameter 30
 - Connector Command Example 31
 - Using Configurations to Group Commands 32
 - Configurations Attributes 34
 - Configurations Contexts 35
 - How to Set Up Command Contexts 36
 - Configurations Commands 37
 - Adding a Command to a Configuration 37
 - Editing Commands in a Configuration 38

Removing Commands from a Configuration	38
Configuration Targets	38
Adding a Target to a Configuration	39
Editing Targets in a Configuration	39
Removing Commands from a Configuration	39
Specifying Targets	39
Target Attributes	40
Target Integration Parameters	40
Authorization and Authentication Settings	41
Setting User Login Parameters	42
Setting Login Credentials	42
Setting Login Credentials on Target Servers	43
Setting Logins and Other Parameters to Prompt for Values at Runtime	43
Access Control Lists (ACLs) on Integration Commands	44
Running Integration Commands	45
Entering/Saving Command Parameters at Runtime	46
Creating New Configurations On-the-Fly	47
Network Tools as Integration Commands	47
Send Documentation Feedback	49

Chapter 1: Overview

Action connectors allow integrations between ArcSight and third party devices; this allows the third party device to be controlled from the ArcSight Console.

You can execute commands on third party devices from within ArcSight and send the output of the commands back to the ArcSight Console. The remote command can be executed as an action in the correlation rules engine, or as a right-click on the connector. The command executes from the host where the connector resides.

This added functionality leads to more cost effective operations as users no longer have to KVM between monitors or switch between detection and action for resolution of events. Not having to leave the ArcSight Console to make changes or to take action is a powerful solution for our joint customers.

Action connectors can interface with Ecosystem partner's software or devices through a command-line-interface (CLI), launching a URL from within ArcSight and, in the future, possibly through Web services or other means.

Once CEF certification is achieved, opportunities to create an Action Connector should be explored. If an opportunity exists, partners can create, test, and have their Action Connector certified by ArcSight. The result is a documented and packaged solution that easily can be made available to customers.

Action Connector Development Options

There are two options for developing Action Connectors. The first is through our Flex CounterACT technology which is documented starting with [Installing the Flex CounterACT Connector](#). The second option is developing an Action Connector through ArcSight's Integration Commands technology which is documented in [Integration Commands](#).

Flex CounterACT

- The Flex CounterACT Connector is included in the SmartConnectors package.
- Actions that are supported by an Action Connector can be automated by actions in the rules engine. Actions can also be initiated by right-clicking on the connector in the ArcSight Console.
- The Flex CounterACT Connector is limited to CLI integrations.
- Commands are executed on the host where the SmartConnector is installed.

Integration Commands

- Commands are executed on the host where the ArcSight Console is installed. Depending on the action being executed, this may require additional configuration that host. For

example, if a Perl script is to be executed, Perl must be installed on each host that will execute the script.

- Actions are initiated using manual “right-click”. Actions cannot be automated by using actions in the rules engine.
- There are more options to integrate with 3rd party devices outside of a CLI.
- Allows for a third-party Web interface to be opened from within the ArcSight Console.

Chapter 2: Installing the Flex CounterACT Connector

Before you start the installation, make sure that ESM is installed correctly. Also make sure you have local access to the machine where the Flex CounterACT Connector is to be installed and that you have the administrator passwords.



Note: The Flex CounterAct Connector is available for installation in one single ESM destination, from which you can execute Flex CounterAct commands.

1. Start the ArcSight SmartConnector Installer by running the SmartConnector executable. Follow the installation wizard through the following folder selection tasks and installation of the core connector software:
 - Introduction
 - Choose Install Folder
 - Choose Install Set
 - Choose Shortcut Folder
 - Pre-Installation Summary Installing...
2. When the installation of ArcSight SmartConnector core component software is finished, the destination selection window is displayed. Choose the ArcSight Manager as the destination.
3. The Wizard first prompts you for Manager certificate information. The default selection is **No**, the ArcSight Manager is not using a demo certificate. Choose **Yes** if ArcSight Manager is using a demo certificate. (Before selecting this option, make sure the Manager is, in fact, using a demo SSL certificate. If you are not certain, select **No** or consult your system administrator.)

If your ArcSight Manager is using a self-signed or CA-signed SSL certificate, select **No**, the ArcSight Manager is not using a demo certificate and click **Next**.

After completing the SmartConnector installation wizard, remember to manually configure the connector for the type of SSL certificate your ESM Manager is using. See the [ESM Administrator's Guide](#) for complete information.
4. The Wizard prompts for **Manager Host Name** and **Manager Port**. Enter the information and click **Next**.
5. Enter a valid ArcSight **User Name** and **Password**. This is the same user name and password you created during the ArcSight Manager installation.
6. At this point, exit the wizard.

7. The Flex CounterACT connector is currently marked as INTERNAL so to install it you need to execute agent setup with the following command line parameters:
`arcsight agentsetup -w -sa`
8. Select the Flex CounterACT Connector from the list of available connectors.
9. Click **Next**.
10. Enter the name of the Configuration File (the extension is added automatically).
11. Complete the remaining steps.

Chapter 3: Flex CounterACT Connector Properties File

The Flex CounterACT Connector uses the properties file to store the commands that will be executed. Create the file `<file_name>.counteract.properties` in the directory `<ArcSight_Home>\current\user\agent\flexagent`. Following is an example of a properties file:

```
command.count=1 command[0].name=quarantine
command[0].displayname=Quarantine

command[0].parameter.count=1 command[0].parameter[0].name=ip command
[0].parameter[0].displayname=Ip

command[0].action="${_ARCSIGHT_HOME}/bin/agent/nrm/${_PLATFORM}/en_ira_cli_
v4${_PLATFORM_BINARY_EXT}"
--action=quarantineNode
--ip=${ip}
--length=0
--motion=add
--overwrite=1
```

The following section provides an explanation of each property:

- `command.count`—The number of commands that are supported by this CounterACT Connector.
- `command[x].name`—The internal name that you want for the command. Typically, the internal name should be all lowercase and contain no spaces.
- `command[x].displayname`—The name that is displayed in the ArcSight Console for this command. Typically, this name should be capitalized properly and it can contain spaces.
- `command[x].parameter.count`—The number of parameters that the command receives.
- `command[x].parameter[x].name`—The internal name of the parameter, typically all lowercase and no spaces.
- `command[x].parameter[x].displayname`—The display name for the parameter (that is shown in the console). Typically this is capitalized properly.
- `command[x].action`—The command line executable that is run. This property should be provided as a template with variables that will be replaced by the actual values. These variables are provided by default:
 - `ARCSIGHT_HOME`—The absolute path to where the connector is running.
 - `PLATFORM`—A platform code (such as `win32/linux/solaris`). Typically, used if you have scripts for different operating systems.

Chapter 4: Flex CounterACT Connector Examples

This section provides examples of how to implement Flex CounterACT Connectors.

- [Nmap Example](#)
- [Advanced Nmap Example](#)

Nmap Example

This example uses the Flex CounterACT Connector to execute Nmap. The properties file looks like this:

```
command.count=1
command[0].name=nmapit
command[0].displayname=NMap

command[0].parameter.count=1
command[0].parameter[0].name=ipaddress
command[0].parameter[0].displayname=Ip Address

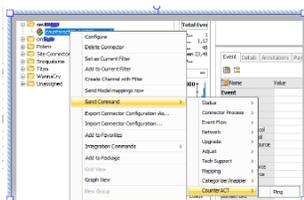
command[0].action=C:\\NMAP\\NMAP.EXE ${ipaddress}
```

The properties file is saved at:

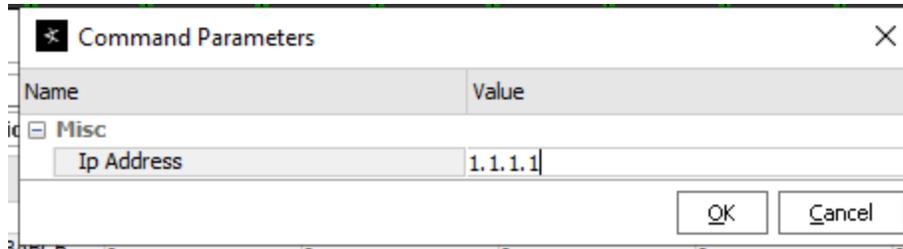
```
{ARCSIGHT_HOME}/user/agent/flexagent/nmap.counteract.properties.
```

From the connector:

Select the connector in the Navigator panel of the ArcSight Console. Right-click and select the **Send Command**. Under **CounterACT**, select the command you want to run.



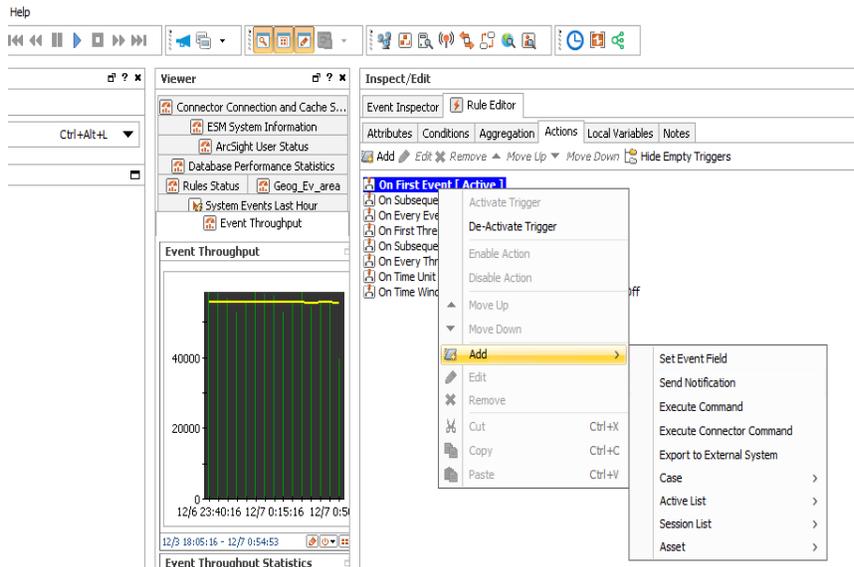
A window will display prompting for any required parameters. Enter the parameters and click **OK**.



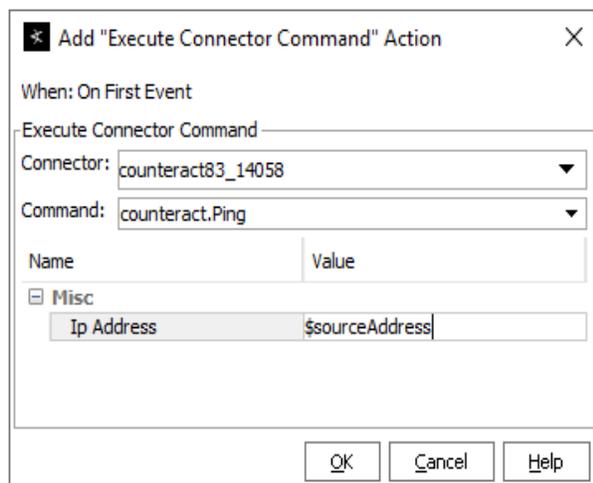
A window will display that shows the output of the command. There will also be an event in the Console with the event name set to Command [counteract.<command name>] executed, the output of the command will be included in the message field.

From a rule:

In the Navigator panel of the ArcSight Console, create or edit a rule. Add an action to the rule and select **Execute Connector Command**.



Select the connector and the command you want to execute from the list of commands. Provide fields, if necessary, for any parameters for the command.



Name	Value
Ip Address	\$sourceAddress

For the Nmap example above, you could enter `$sourceAddress` for the parameter. Whenever this rule fires and triggers the action, it will execute the Nmap command using the source address of the event that triggered the rule.

A window will display that shows the output of the command. There will also be an event in the Console with the event name set to "Command [counteract.<command name>] executed", the output of the command will be included in the message field.



Note: Avoid using commands that "hang" or take too long to complete. The connector has a default timeout of 5 minutes, if the command does not finish in 5 minutes, it will be terminated. It is possible to change the length of the timeout if necessary.

The message field can only hold 1k of data. If your command generates a longer response than that, enable **Preserve Raw Event** in the configuration of the connector in the Console. The output of the command will also be present in the `rawEvent` field. The raw event has a capacity of 4k of data.

Advanced Nmap Example

It is possible to do processing on the result of the command as well. For example, to run Nmap to check if the server has port 22 open (SSH). The command to do that would be:

```
nmap 1.1.1.1 -p22
```

If the port is open, the output would be similar to:

```
Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2007-04-18 21:43 PDT
Interesting ports on somehost.sv.arcsight.com (1.1.1.1): PORT STATE SERVICE
22/tcp open  ssh
```

```
Nmap finished: 1 IP address (1 host up) scanned in 0.132 seconds
```

If the port is closed, the output would be similar to:

```
Starting nmap 3.81 (http://www.insecure.org/nmap/ ) at 2007-04-18 21:43 PDT
```

```
Interesting ports on somehost.sv.arcsight.com (1.1.1.1): PORT STATE SERVICE
22/tcp closed ssh
```

```
Nmap finished: 1 IP address (1 host up) scanned in 3.137 seconds
```

It is possible to parse the output and modify the return event to extract the STATE of the port you are looking for using a module called `SecondLevelRegexParser`. An example of how to parse the output of the Nmap command:

The properties file (`user/agent/flexagent/nmap.counteract.properties`) would be:

```
command.count=1
command[0].name=nmapit
command[0].displayname=SSH probe
command[0].parameter.count=1
command[0].parameter[0].name=ipaddress
command[0].parameter[0].displayname=Ip Address
command[0].action=nmap ${ipaddress} -p 22
```

Note that the command will now be displayed as SSH probe which makes it clear for the ArcSight Console user.

To use the second level parser feature, create the file `user/agent/fcp/additionalregexparsing/ngflexcounteract/regex.0.sdkrfilereader.properties` with the following content:

```
source.field=event.rawEvent
regex=(?s).*?22/tcp\s+(\S+).*
token.count=1
token[0].name=PortState
event.name=__concatenate("SSH is ",PortState)
```

The `source.field` tells the parser that the value of `event.rawEvent` is the value that is being parsed. Remember that the `rawEvent` field will contain the output of Nmap command.

The expression (`regex`) looks for `22/tcp (\S+)` which will match the STATE of the port. The `(?s)` at the beginning of the expression means that the expression is a MULTILINE expression which means that the dot (`.`) matches new line, so the `.*` effectively matches complete lines before the `22/tcp`.

You can write multiple second level regex parsers if needed. For example, to add the IP address that was scanned into `destinationAddress`, you could create an additional file

user/agent/fcp/additionalregexparsing/ngflexcounteract/regex.1.sdkrfileloader.properties with the following content:

```
source.field=event.rawEvent
regex=(?s).*?Interesting.*?(\\S+) \\((\\S+)\\).*
token.count=2
token[0].name=MyHostName
token[1].name=MyAddress
token[1].type=IPAddress
event.destinationAddress=MyAddress
event.destinationHostName=MyHostName
```

The second level regex parsers are regular expression FlexConnector files, so all features including submessages are supported (see the *FlexConnector Developer's Guide* for more information).

Restart the connector to load these files.

To make sure your second level regex parser file is being parsed, look for the following message in logs/agent.log :

```
[2011-04-18 22:09:28,743][INFO][default.com.arcsight.agent.loadable._
SecondLevelRegexParser][processSingleAlert] Loading second level regex parser
for agent type [ngflexcounteract] from
[additionalregexparsing/ngflexcounteract/regex.0]
[2011-04-18 22:09:28,743][INFO][default.com.arcsight.agent.loadable._
SecondLevelRegexParser][processSingleAlert] Loading second level regex parser
for agent type [ngflexcounteract] from
[additionalregexparsing/ngflexcounteract/regex.1]
```

Chapter 5: Integration Commands

This chapter contains the following information:

- [What are Integration Commands?](#)
- [Planning Checklist and Workflow](#)
- [Navigating to Integration Command Resources](#)
- [Quick Example](#)
- [Defining Commands](#)
- [Using Configurations to Group Commands](#)
- [Specifying Targets](#)
- [Authorization and Authentication Settings](#)
- [Running Integration Commands](#)
- [Network Tools as Integration Commands](#)

Integration commands leverage the power of security and event management, and broaden its view to show external, snap-in views from appliances like ArcSight Logger, as well as third-party applications.

ArcSight ships with standard content (pre-built commands) and a platform for building your own command configurations.

Contact ArcSight Professional Services if you need assistance in authoring tools integrations with ArcSight products or other applications.

What are Integration Commands?

Integration commands enable you to link from the ArcSight Console to information in other views and applications. You can also build and launch commands locally and on remote servers or appliances, using field values in events as command parameters. You can configure the commands as context-aware, right-click options on different views, resources, and editors on the ArcSight Console.

Configurations can define valid data types and selections for a set of commands. For example, you could configure a set of URL commands to run as a right-click on a selected cell in an active channel and accept only IP addresses as data types.

The ability to integrate commands for various applications means the ArcSight Console can serve as a central hub for defining, managing, and launching actions, Logger searches, and third party applications, as well as local ArcSight scripts. You can also configure and manage role permissions and access lists (ACLs) for tools and commands in the ArcSight Console.

Supported Command Types

You can build these types of context-based, right-click commands into the ArcSight Console:

Command Type	Output Results	Examples
URL commands link to Web page URLs or URIs	<ul style="list-style-type: none">• ArcSight Console internal browser (Windows only)• External Web browser	<ul style="list-style-type: none">• Out-of-the-Box Logger Searches• Out of the box URL commands
Script commands run scripts	Script/executable output result	Network Tools
Connector commands are derived from associated nodes or applications	Structured result based on the SmartConnector and its associated node or application	Connector commands

For more information on working with commands, see [Defining Commands](#).



Tip: All integration commands are designed as manual, right-click options in various contexts in the ArcSight Console. This enables you to launch commands in ArcSight Console displays and, access available work flows in other applications.

To define rule-driven commands, configure rule actions to send SmartConnector commands (rather than by creating integration commands).

Out-of-the-Box Commands for Logger

ArcSight ships with pre-built, URL-based commands for the ArcSight Logger. A typical command would be to run a remote search or query on an element in a selected Logger stored event in an active channel.

Local Scripts and Commands to Other Applications

Typical activities for which you might build and run commands in the ArcSight Console that connect to other applications and tools include:

- Launch third-party Web interfaces
- Launch scripts
- Run external searches
- View submitted tickets
- Get Asset/Vulnerability information
- Get Payload Information

You can set up context-aware commands to third-party applications and custom scripts. With command configurations, you can make these available in specified ArcSight Console views and use particular fields as parameters to your commands.

ArcSight ships with standard utilities configured to be available in ArcSight Console views. For example, the **ping** command is available in grid views such as active channels, lists, and query viewers, and takes as a parameter the IP address or host name in the selected event.

For information on integrating basic network tools such as Ping, Nslookup, or ArcSight specific “Send Logs”, see the information about using network tools in the [ArcSight Console User's Guide](#) and [Network Tools as Integration Commands](#).

How it Works

Integration commands provide resources for tools integration authors to:

- Build context-sensitive commands that you can run locally or on multiple, remote target servers, and you can mix, match, and re-use with configurations.
- Associate parameters with commands to read the resources for which you call the commands. Command parameters make use of Velocity Expressions to pick up values from fields and resources. (See the information about Velocity templates in the [ArcSight Console User's Guide](#).)
- Define configurations sets of commands) for various external applications to specify relevant contexts, commands, rendering formats, and, optionally, remote targets.

Once integration commands and configurations are in place, analysts and operators working with the ArcSight Console can use your custom-built commands or ArcSight pre- built commands (for Logger) to manage and monitor networks and assets with an extended reach into other views, toolkits, and servers.

Configure Login credentials for authentication on external applications through integration parameters on the user resource. See [Setting User Login Parameters](#) and [Setting Logins and Other Parameters to Prompt for Values at Runtime](#).

Define authorization to use or edit commands through access control lists (ACLs) as described in [Access Control Lists \(ACLs\) on Integration Commands](#).

Planning Checklist and Workflow

Plan your command integrations by identifying the utilities or applications to integrate and collecting the necessary information. Here is a checklist of considerations.

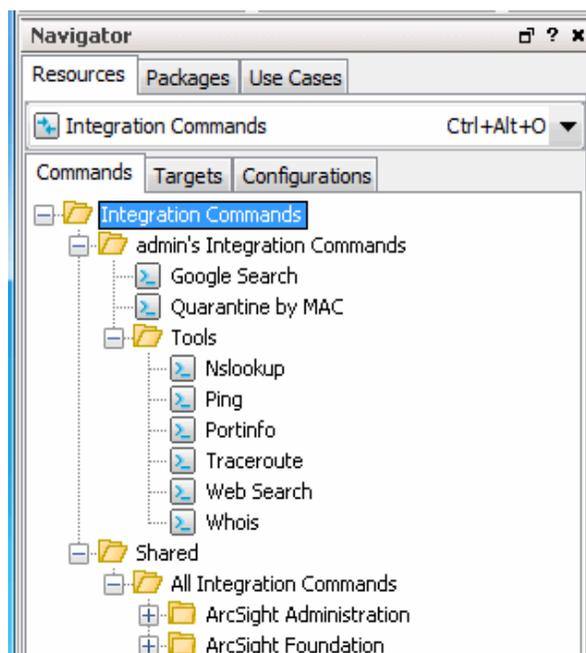
Components	Questions
Commands	<ul style="list-style-type: none">• What commands will you run on the external application? Is there a subset of commands you want to integrate into the ArcSight Console?• What is the command type (Web URL, local executable script, or Connector command) and syntax.
Servers, Authentication SmartConnectors	<p>Integrating Logger URL commands requires an IP address or Host name of the appliance and authentication credentials for users.</p> <p>Integrating a Connector command requires access to the appliance, and a connector deployed and registered with the Manager to which your ArcSight Console is connected.</p>
Configuration	<ul style="list-style-type: none">• How do you want to render (display) output results of commands? This largely depends on the command type; for example, URL commands are rendered in an external or embedded browser.• How many integration configurations do you need?• Does the application you are integrating have more than one type of interface? (for example, Web and CLI) If so, you'll need a configuration for each interface and associated command type.
Users	<ul style="list-style-type: none">• Which users work with these integration tools or applications?• Are authentication parameters required on target servers, appliances, or applications? If so, collect or establish user names and passwords for users who run these commands.• Plan for configuring integration parameters on user accounts for users who work with the external applications. These users need login credentials for both ArcSight and the target applications.• For users with the same authentication parameters for a target server, you can create a target resource with those parameters instead of duplicating the parameters in each user account. Then you can configure the ACL of that target resource so that only those users have access to it. When a command is triggered in the right context, only the target to which that user has access is displayed. Use a similar ACL approach for commands. For example, a single configuration can contain groups of commands, where some commands require special privileges.

Once you have a plan, you might try configuring the commands and testing in this order:

1. Add the commands (command name, type, the command itself, and its parameters).
2. Specify the targets (remote servers where commands run), if any.
3. Create one or more configuration(s), add in the commands you created, choose how command results are rendered (displayed), and define ArcSight Console UI contexts where these commands are available for use.
4. Add Integration Parameters to User Accounts. If authentication is required on target servers, configure login credentials on user accounts for users who run these commands. These users need login credentials for both ArcSight and on the target applications.
5. Test the commands. See [Running Integration Commands](#).

Navigating to Integration Command Resources

To create or edit integration commands and configurations, start by navigating to **Integration Commands** resources.



Users can access existing integration commands and configurations through right-click commands on the ArcSight Console in various contexts. The contexts depend on how the commands are configured.

Quick Example

To experiment with building integration commands, you need one command and one configuration. Create the command(s) first because the configuration references the commands.

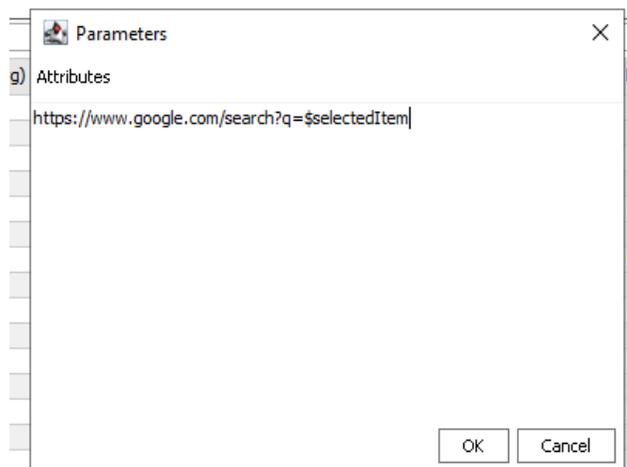
The configuration also defines how command results are rendered, and references contexts where your new Integration Commands appear in the ArcSight Console right-click menus (for example, Viewers, Resource Panel, Editors, and more specifics within those contexts).

To define targets (remote servers where commands run), add them to the configuration.

Here is an example of how to set up a command to do a Google Search on a selected cell in the ArcSight Console. This example does not require a "target," so just set up a command, add it to

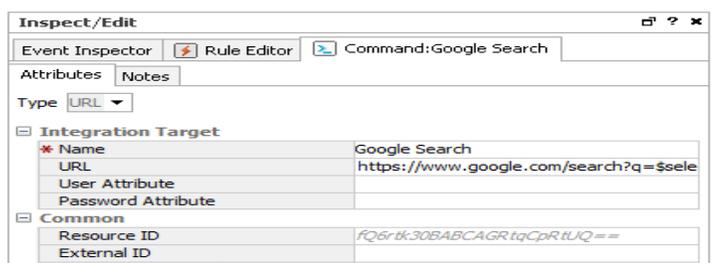
a configuration, and run it. The details of this and other types of commands and configurations are discussed further in the topics that follow.

1. Start by getting the format of the Google search. Do a Google Search in a Web browser. Copy the first part of the URL (everything before or to the left of the search term) from the Address bar, so you have it on your clipboard. (You will be using this to paste in to the Parameters dialog in [Step 4](#).)
2. Now let's set up the command. In the ArcSight Console Navigator panel, select the **Integration Commands** resource from the drop-down menu and click the **Commands** tab.
3. Right-click the group (folder) where you want to create the command and select **New Command**.
4. On the Commands Editor, fill in these attributes:
 - For command **Type**, choose **URL**.
 - For **Name**, provide a user-friendly name like "Google Search".
 - For **URL**, click the browse button to get the Parameters dialog. Paste the Google search prefix (from [Step 1](#)) into the Parameters dialog scratch pad:
`http://www.google.com/search?q=`
 - Click **Attributes** on the Parameters dialog to get a list of Velocity Expressions. Select the option, **Selections** > `$selectedItem`. The expression is added as a parameter to the search:
`http://www.google.com/search?q=$selectedItem`



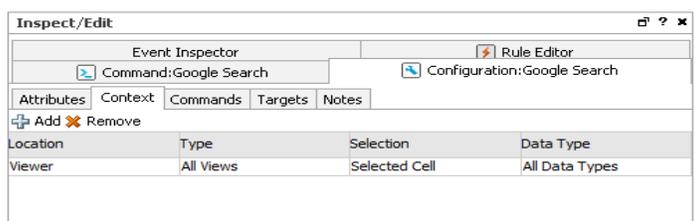
- Click **OK** to close the Parameters dialog and save your changes.

- Click **Apply** or **OK** on the Commands editor to save the command.



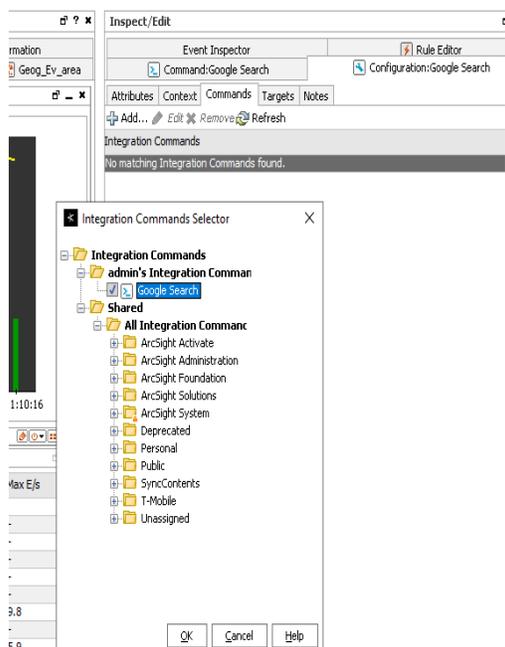
5. Now, let's set up the configuration and add the command to it. Click the **Configurations** tab.
6. Right-click a group and select **New Configuration**.
7. On the Configurations Editor, select **URL** as the configuration Type, and provide a user-friendly name.
8. Still on the Configurations Editor, click the **Context** tab. This sets where in the ArcSight Console the command is available. Click **Add** to get a set of context fields, then click into each field to select a location, type, selection, and data type. (You can add multiple contexts by clicking Add again.) Add one context to show in the Viewer in all "views" and to take the selected cell as the "selection":

Location	Type	Selection	Data Type
Viewer	All Views	Selected Cell	All Data Types



When the search command is deployed as part of this configuration, and run via a right-click command in the context of the ArcSight Console, it searches on the text in the "cell" (Viewer table cell) the user selects in the ArcSight Console.

9. Finally, add the command to the configuration. On the Configuration Editor, click **Commands**. Click **Add** to get the command selector, select your Google Search command, and click **OK**.



10. Click **Apply** or **OK** on the Configurations Editor to save the configuration.

Now run the Search command you just built

11. Open any active channel, list, data monitor, or query viewer with a table style view.
12. Right-click any cell in the Viewer that contains a term you would like to search on, and select **Integration Commands > Google Search** (or whatever you named the command).

The command runs a search using the text from the selected cell as the search term, and returns search hits in the browser (on 32-bit Windows, either the ArcSight Console internal browser or an external Web browser.)

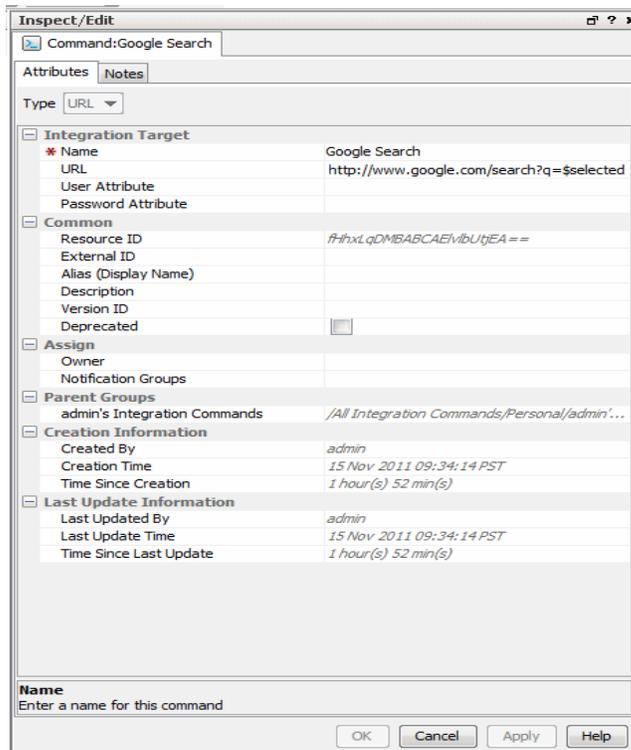
The following topics provide more information and examples on how to build all types of commands, how to add user authentication, how to use targets for commands, how the standard network tools are implemented as integration commands, and more.

Defining Commands

Use the commands feature to configure URL, Script, and Connector commands for custom and third party applications and other ArcSight products. Setting up commands is the first step in a multi-part process to providing a set of integration commands. (Other tasks include setting up configurations, targets, and user login parameters). This topic explains how to add and edit the command portion of an integration command solution.

To add a new command, do the following:

1. In the Navigator panel, select the **Integration Commands** resource from the drop-down menu and click the **Commands** tab.
2. Right-click a group (folder) in which to create the command, and select **New Command**. This launches the Command Editor in the Inspect/Edit panel. It is best to create new content in your own folder.
3. On the **Command Editor**, select the command **Type** and fill in the fields for command Name, and other attributes.



Command Type	Description
Script	Executable script that runs locally to the ArcSight Console where the command is launched.
URL	Web URL for which you can define parameters.
Connector	Commands for ArcSight Connectors.

4. Click **Apply** or **OK** to add the new command.

Command Types and Attributes

The command attributes vary, depending on the type (Script, URL, or Connector), as described below.

Script Commands

Like other commands, you can make script commands available to multiple users and user groups. Users probably run the ArcSight Console on many different machines. Integration script commands always run on the same machine as the ArcSight Console used to launch them. Therefore, the working directory and program path names should reflect where commands are found in ArcSight Console users' environments

Attribute	Description
Name	User-friendly Name for the command.
Working Directory	Directory containing the executable script. For example, <code>\$systemRoot\system32\</code> You can type the directory path in the Program field, or click the Browse Directory button  to get a file browser. Use the file browser to navigate to and select the command. Note: Be sure this path reflects the location of the script on machines used by ArcSight Console users for whom you are building these commands.
Program	Full path to the executable command. For example, <code>\$systemRoot\system32\ping.exe</code> You can type the full path to the command in the Program field, or click the Browse Directory button  to get a file browser. Use the file browser to navigate to and select the command. Note: Be sure this path reflects the location of the script on machines used by Console users for whom you are building these commands.
Parameters	Provide parameters for the command. (See Adding and Editing Command Parameters) The Attributes list provides Velocity Expressions for all event fields and an option to add <code>\$selectedItem</code> as an attribute.



Tip: Entering data in the Common and Assign sections is optional, depending on how your environment is configured. For information about the Common and Assign attributes sections, as well as the read-only attribute fields in Parent Groups and Creation Information, see the information about common resource attribute fields in the [ArcSight Console User's Guide](#).

URL Commands

Attribute	Description
Name	User-friendly name for the command.
URL	<p>The URL for the command, along with any parameters provided as arguments to the URL.</p> <p>Click the browse button  to get the Parameters dialog. (See Adding and Editing Command Parameters for information on how to add the URL along with parameters or arguments to the URL.) You can copy/paste URLs onto the Parameters dialog scratch pad or type them directly. The Attributes link provides Velocity Expressions you can add as parameters (attributes) to the URL.</p> <ul style="list-style-type: none">• Type or paste URL directly in the Parameters dialog scratch pad.• Click Attributes to add a Velocity Expression as a URL parameter. <p>Determine the URL by first accessing it from a Web browser address bar. This also shows you where in the URL to add the parameters (if any).</p> <p>Example: Web Search</p> <p>To set up a Google Search on a parameter, do a Google Search in a Web browser. Extract the first part of the URL (everything to the left of the search term) from the Address bar, and paste it into the Parameters dialog scratch pad: <code>http://www.google.com/search?q=</code></p> <p>Click Attributes on the Parameters dialog to get a list of Velocity Expressions. Select the option, Selections > \$selectedItem. The expression is added as a parameter to the search:</p> <pre>http://www.google.com/search?q=\$selectedItem</pre> <p>Click OK to close the Parameters dialog and save your changes. Click Apply or OK on the Command Editor when you are satisfied with all settings.)</p> <p>When this search command is deployed as part of an integration configuration, and run via a right-click command in the context of the ArcSight Console, it searches the text in the cell (Viewer table cell) the user selects in the ArcSight Console.</p>
Parameters	Parameters for URL commands are added as attributes to the URL as described above in URL .



Tip: Entering data in the Common and Assign sections is optional, depending on how your environment is configured. For information about the Common and Assign attributes sections, as well as the read-only attribute fields in Parent Groups and Creation Information, see the information about common resource attribute fields in the [ArcSight Console User's Guide](#).

Connector Commands



Note: Prerequisites for Connector Commands

If you plan to build and use Connector commands you need:

- One or more of the associated SmartConnectors deployed and registered with the Manager to which your ArcSight Console is connected.

Test connectivity and authentication between your local machine, SmartConnector(s), and appliance(s) before setting up Connector integration commands.

Attribute	Description
Name	User-friendly Name for the command.
Group	Choose a group from the Group drop-down menu. Depending on which Group you select, relevant commands are provided in the next field (Command). See the information about executing connector commands in the ArcSight Console User's Guide .
Command	Choose a command from the drop-down menu. Depending on which Group you selected, relevant commands are provided here. Choose a Connector command from the drop-down list. Note: To get the list of Connector commands, you must have the SmartConnector deployed and registered with the Manager to which your ArcSight Console is connected.
Parameters	To define parameters for the command: <ol style="list-style-type: none">1. Click the browse button  to get the Parameters dialog. A table of name-value pairs is provided that represents the valid parameters for the given command.2. Select the parameters to use, and define values for them with either hard-coded values or Velocity Expressions. For example, you could define the Connector command Quarantine Node By IP Address to use three parameters; IP Address, Quarantine Period, and Overwrite Active Quarantine (a yes/no value set to 0 or 1, respectively). You could set the IP address to a Velocity Expression for attacker address, Quarantine Period could be set to 1 hour, and overwrite set to Yes. The Attributes list provides Velocity Expressions for all event fields along with options to add Console selections, dates, and channel start and end times as attributes.3. Click OK on the Parameters dialog to save your changes.



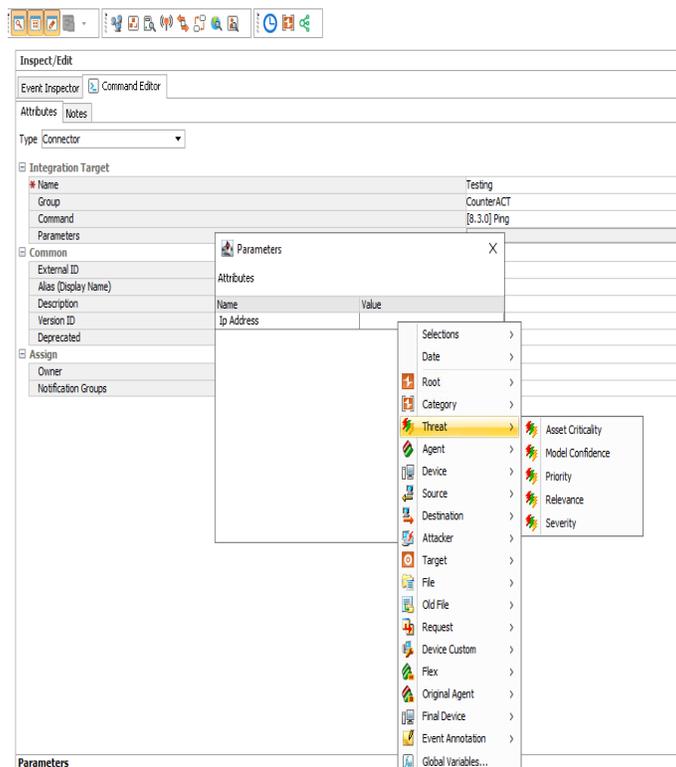
Tip: Entering data in the Common and Assign sections is optional, depending on how your environment is configured. For information about the Common and Assign attributes sections, as well as the read-only attribute fields in Parent Groups and Creation Information, see the information about common resource attribute fields in the [ArcSight Console User's Guide](#).

Adding and Editing Command Parameters

The Attributes list includes Velocity Expressions for all event fields and an option to add user field or item selections, channel start or end time, date/time, and other Velocity Expressions as attributes.

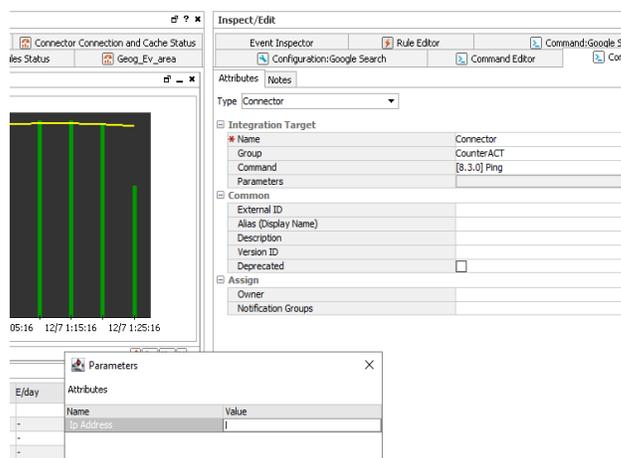
To provide Parameters for a command:

1. Click the browse button  to get the Parameters dialog.
2. Click **Attributes** to get a list of variables and Velocity Expressions.



3. Select the expression you want to add. The attribute list includes Global Variables. If the global variable to add is composed of a list of fields, expand the global variable displayed in the Parameters dialog and select the field you want.

The expression is added to the Edit Attributes scratch pad as a parameter.



4. You can continue adding expressions, which are chained together.

For example, selecting **Threat > Priority** from the Attributes list results in this parameter being placed on the scratch pad:

```
#{priority}
```

Subsequently selecting **Attacker > Address**, updates the scratch pad entry with chained-together expression:

```
#{priority} #{attackerAddress}
```



Tip: The Parameters dialog is an editable scratch pad

In addition to adding Velocity Expressions from the Attributes menu and Templates for Connector command parameters, you can type new expressions directly into the dialog. Also, you can select and edit existing expressions manually. See also [Removing a Command Parameter](#).

5. When the Parameters scratch pad reflects the expressions you want to include as command parameters, click **OK**.

The parameters you added are reflected on the Attributes tab in the Command Editor.



6. Click **Apply** or **OK** on the Command Editor to save changes to command parameters along with any other changes to the command that you want to retain.

Removing a Command Parameter

To remove a command parameter:

1. Click the browse button to get the Parameters dialog.
2. Select the parameter in the scratch pad and hit the Delete key on your keyboard.
3. To add a new parameter to replace the one you are deleting, do so by following steps described in [“Adding and Editing Command Parameters”](#) on page 28.

4. Click **OK** on the Parameters dialog.
5. Click **Apply** or **OK** on the **Command Editor** to save your changes.

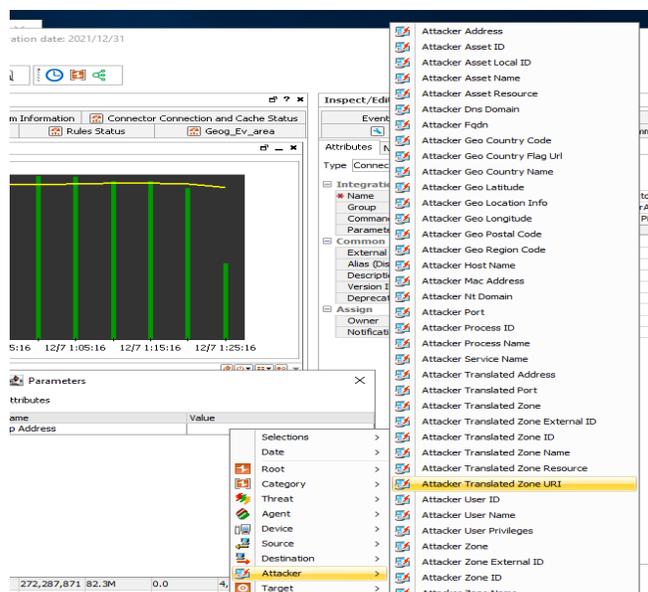
Connector Command Example

Here is an example of creating a Connector command. In this example, you use the command **Quarantine Node By IP Address** to quarantine an attacker node. Use three parameters: IP Address, Quarantine Period, and Overwrite Active Quarantine. In the Parameters field for this command, use hard-coded values for length and overwrite, but use a Velocity Expression to let the user derive the attacker address for a selected event on the ArcSight Console.

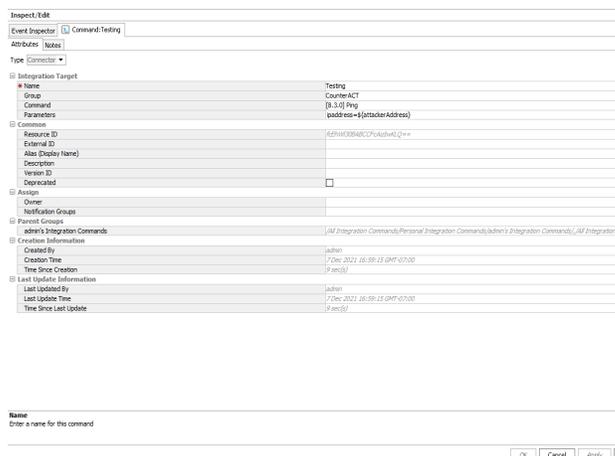
1. Create a new command of type **Connector** and name it.
2. In the Group field, select **Connector**.
3. In the Command drop-down menu, select **Quarantine Node By IP Address**.
4. In the Parameters field, click the browse button  to get the Parameters dialog.

Right-click the Value field next to IP Address on the Parameters dialog to get the Velocity Expression chooser.

To get the Velocity Expression `${attackerAddress}`, select **Attacker > Attacker Address** on the Parameters dialog.



To quarantine the attacker node for 1 hour and overwrite the value, enter the number 1 for both **Quarantine Period** and **Overwrite Active Quarantine**.



Press the **Enter** key after adding each value to be sure it is applied.

After entering the parameter values, the command attributes are defined as follows:

Parameter Name	Value
IP Address	<code>\$\$\$attackerAddress</code>
Quarantine Period	1
Overwrite Active Quarantine	1

Click **OK** to save them.

- Review the parameter values on the Editor Attributes tab.

`ip=$$$attackerAddress,length=1,troubleticket=,comment=,overwrite=1`

You might have to stretch the Editor window to see all the parameters.



- Click **Apply** or **OK** to save the command.

Using Configurations to Group Commands

An integration configuration resource represents a family of commands of the same type. Commands in a configuration share the same context, rendering method, and targets.

Configurations provide a way of grouping similar commands and specifying common options for where on the ArcSight Console UI the commands are available (contexts), how command results are displayed, and where commands run (scripts run locally; others, like Connector commands, can have one or more remote targets). This is partly a matter of preference (about how you want to group, organize, and present commands to ArcSight Console users), and partly a matter of which commands belong together.

Typically, each integration maps to a single product. However, you can distribute sets of commands across multiple configurations, if needed. This is useful when the same product has different types of interfaces.



Note: Configurations can include only commands of the same type (script, URL, or Connector)

Commands that share a configuration use the same contexts, and (if relevant) targets. You might want to make finer-grained groupings; for example, sub-groups of scripts or Connector commands.

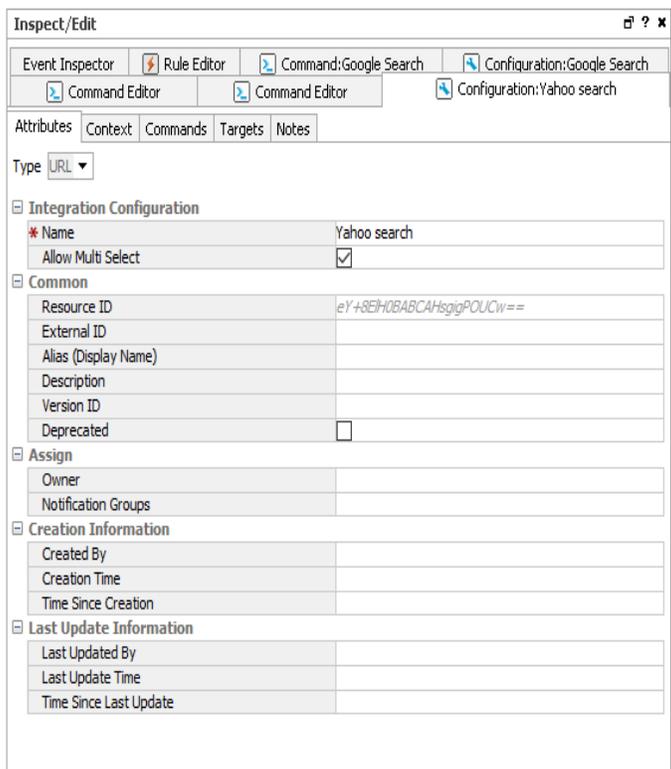
For example, you might group a set of commands that deal with quarantine of nodes into a single configuration. Or you might group a set of URL-based commands used for searching and researching on particular types of events (via Google Searches, Knowledge Base articles, and so forth).

Setting up configurations is a step in a multi-part process of making a set of integration commands available to ArcSight Console users. (Other tasks include setting up commands, targets, and user login parameters).

This topic explains how to add and edit the configuration portion of an integration command solution. For an overview of the integration commands feature, see [Integration Commands](#). For more details on the relationship between commands, configurations, and targets, see [How it Works](#).

To create a configuration:

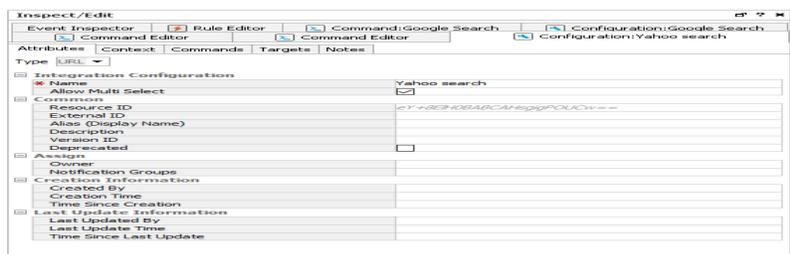
1. In the Navigator panel, select the **Integration Commands** resource from the drop-down menu and click the **Configurations** tab.
2. Right-click a group (folder) where you want to create the configuration, and select **New Configuration**. This launches the Configurations Editor in the Inspect/Edit panel.
3. Fill in the fields on Attributes, Context, Commands, and Targets tabs as described in:
 - [Configurations Attributes](#)
 - [Configurations Contexts](#)
 - [Configurations Commands](#)
 - [Configuration Targets](#) (when commands run on remote targets)



4. Click **Apply** or **OK** to add the new configuration.

Configurations Attributes

Define the configuration name and other basic details for the configuration on the **Configurations>Attributes** tab.



Attribute	Description
Type	<p>Choose the type of configuration from the drop-down menu:</p> <ul style="list-style-type: none">• Script• URL• Connector <p>Note: The configuration type must match the command types in the configuration. See Defining Commands. Once the configuration is saved, the type is not editable. This setting influences choices on other options for the configuration.</p>
Name	<p>A user-friendly, informative name for the configuration that (preferably, one that indicates the commands contained in it).</p>
Allow Multi Select	<p>Use this to allow selecting multiple events on which to run a command. It is off by default. A check mark indicates it is on/enabled.</p> <p>When on, users can select multiple events and the commands assign the values to a parameter as a comma-separated list.</p> <p>For example, suppose you have a command with the parameter <code>ip=\$targetAddress</code>.</p> <ul style="list-style-type: none">• With Multi Select disabled, the command accepts only a single IP address based on a selected event (for example, <code>ip=127.1.0.0</code>).• With Multi Select enabled, a user can also get <code>ip=127.1.0.0,192.168.1.1</code> if two rows are selected. <p>For this to work: (1) the ArcSight Console context (for example, active channel) must allow multi-row selection, and (2) the integration target must support a comma-separated list of values for the given command and parameter.</p> <p>Note: Multi Select does not affect how individual fields in an event are processed. Event field processing is determined entirely by the definition of command parameters. For example, a command with an Attacker Address parameter always gets that value from the selected event.</p>

 **Tip:** Entering data in the Common and Assign sections is optional, depending on how your environment is configured. For information about the Common and Assign attributes sections, as well as the read-only attribute fields in Parent Groups and Creation Information, see the information about common resource attribute fields in the [ArcSight Console User's Guide](#).

Configurations Contexts

As a part of constructing command configurations, you can configure contexts for where in the ArcSight Console certain commands are available. At the same time, you can define parameters for picking up and passing the value in any selected cell, row, or event field.

For example, you could configure a URL command for a Google search as a right-click command on any cell in an ArcSight Console grid view. By using a parameter as the argument to the search command, you could pick up the text from the selected cell or value from any selected field to use as your search term. (In the Commands editor, all fields, provided as a list of Velocity Expressions, are available for use as command parameters.)

Once configured, integration commands are available on right-click context menus from a variety of contexts including:

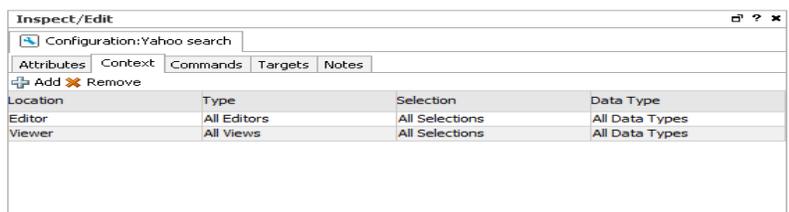
- Relevant fields in active channels (for example, IP address, host name, MAC address)
- Relevant resources (for example, assets)
- Active Lists, sessions lists, query viewers and channels

Also, you can configure **user login parameters** on ArcSight Console users (by using the Integration Parameters tab in the Users resource editor), thereby binding user login information to commands for third-party or ArcSight applications that require secure logins. See [Setting User Login Parameters](#).

You can configure a command to prompt for parameter information, which is often useful for login scenarios and as well as others. See [Setting Logins and Other Parameters to Prompt for Values at Runtime](#).

How to Set Up Command Contexts

Use controls on the Configurations **Context** tab to add, edit, or remove contexts in a configuration.



Click the fields under Location, Type, Select, and Data Type to get drop-down menus with which to select contexts in the ArcSight Console UI where the command is available and to which selections it applies.

Attribute	Description
Location	View where in the ArcSight Console the command is available. For example: <ul style="list-style-type: none">• Viewer, for the Viewer panel where Views of active channels, dashboards, and so on are shown• Resource, for the Navigator Panel resource tree• Editor, for resource editors
Type	Contexts in the ArcSight Console panels where the command is available. Available types vary depending on the location you choose. For example, if you choose Viewer for the location, you can specify types of “views” where you want the command to display, such as Grid View, Chart View, various List entries, Dashboards, Query Viewers, and so on.

Attribute	Description
Selection	User selection or subset of it that is fed into the command. Options can include All Selections, Selected Cell, Selected Row, and Selected Attribute.
DataType	Data type for the parameters fed into the command (derived from the Selection). Options include: <ul style="list-style-type: none">• All Data Types• IP Address• MAC Address• Date• Double• Integer• Long• Resource• String

Configurations Commands

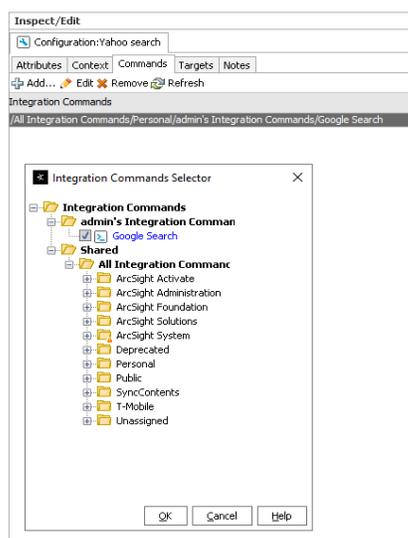
Use controls on the Configurations **Commands** tab to add, edit, or remove commands in a configuration.



Adding a Command to a Configuration

On the Configurations **Commands** tab:

1. Click **Add** to bring up the Commands Selector dialog.



2. Navigate to and click (checkmark) the commands you want to add, and click **OK**.
The commands are added to the list. (You can add multiple commands to a single configuration.)

Editing Commands in a Configuration

On the Configurations **Commands** tab:

- Select the command you want to edit and click **Edit**.

This provides a shortcut into the **Command Editor** for the selected command. See Step 3 in [Defining Commands](#) and [Command Types and Attributes](#) for information on editing the command.

Removing Commands from a Configuration

On the Configurations **Commands** tab, select a command in the list and click **Remove**.

Configuration Targets

Targets are not required for all command types, only for those that run on remote servers. Before you can add a target to a Configuration (explained here), you first need to define it as described in [Specifying Targets](#).

Use controls on the Configurations **Targets** tab to add, edit, or remove targets in a configuration.

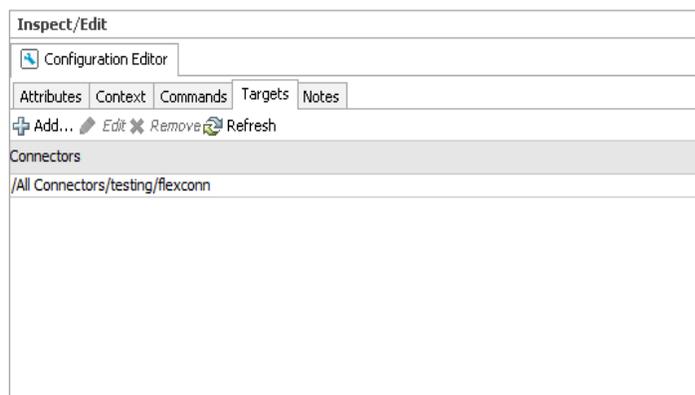


Note: If you plan to add remote targets to a configuration

You need host information for the remote servers and login credentials if authentication is required. See the information about configuring the SmartConnector in the [ArcSight Console User's Guide](#).

Adding a Target to a Configuration

Targets are applicable to Connector commands and any other commands that you want to send to a remote server.



1. Click **Add** to bring up the Connectors Selector dialog.
2. Navigate to and select the target you want to add, then click **OK**.

Editing Targets in a Configuration

1. Select the target you want to edit and click **Edit**.
2. This provides a shortcut into the **SmartConnector Configuration** Editor for the selected connector or target. For more information, see the information about configuring SmartConnectors in the [ArcSight Console User's Guide](#) and also the *SmartConnector User's Guide*.

Removing Commands from a Configuration

On the Configurations **Contexts** tab, select a target in the list and click **Remove**.

Specifying Targets

Optionally, you can specify targets (remote servers where one or more commands run).

If you have multiple remote servers, you might want to configure multiple targets on which to run a single command with the same or different parameters.

For example, you can configure any of the following as command targets.

- Applications with Web interfaces/clients like
 - ArcSight Logger appliances
 - Search providers (for example, Google, Yahoo, ask.com)
 - IT/Security portals
 - Asset/Vulnerability information
 - Ticketing Web servers
- Connectors

Setting up targets is a step in a multi-part process of making a set of integration commands available to ArcSight Console users. (Other tasks include setting up commands, configurations, and user login parameters).

This topic describes how to add and edit the configuration portion of an integration command solution. For an overview of the integration commands feature, see [Integration Commands](#).

To add a new target, do the following:

1. In the Navigator panel, select the **Integration Commands** resource from the drop-down menu and click the **Targets** tab.
2. Right-click a group (folder) where you want to create the target, and select **New Target**. This launches the Command Editor in the Inspect/Edit panel.
3. Fill in the fields as described below.
4. Click **Apply** or **OK** to add the new target.

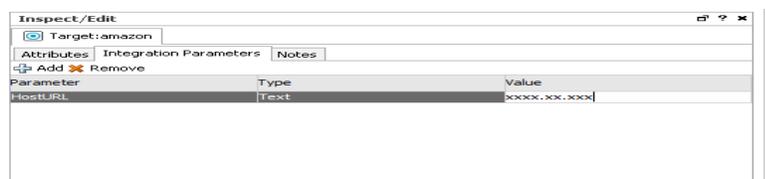
Target Attributes

The only target attribute you need to provide is a user-friendly name for the server.

Attribute	Description
Name	Name for the remote server or appliance where the command run.

Target Integration Parameters

Targets are used only for URL configurations, where you parameterize the Web host target of the URL, and sometimes login credentials. Type directly into the fields to define a parameter, as described below.



Field	Description
Parameter	Parameter name, as specified in the command definition related to this target.
Type	Parameter type. Choose Text or Password from the drop-down menu. Password type parameters are automatically encrypted. Note: <ul style="list-style-type: none">• Always set login credentials (passwords or authentication tokens) to type “Password” (not “Text”). (Credentials set to “Text” are not masked on the UI.)• You can set passwords and authentication credentials on target servers too, but we recommend against it in most cases. Doing so risks opening up a target server to any user who has access to the integration commands (not necessarily an account on the target server). Additionally, it does not give you any tracking information based on user logins to the server.
Value	Hard-coded value, variable, or Velocity Expression for the parameter. For example: <ul style="list-style-type: none">• A host name or IP address as a value for a target server parameter

To add a new parameter, click **Add**. This gives you a new row in which to enter Parameter, Type, and Value information. You can add multiple parameters to a target.

 **Tip:** Entering data in the Common and Assign sections is optional, depending on how your environment is configured. For information about the Common and Assign attributes sections, as well as the read-only attribute fields in Parent Groups and Creation Information, see the information about common resource attribute fields in the [ArcSight Console User's Guide](#).

Authorization and Authentication Settings

Authentication: You can specify user login behavior for commands designed to run on secure, remote target servers. You can specify login credentials to be used as part of the command, or set parameters that prompt users to enter their user name and password when they run the command.

Authorization: You can set up fine-grained access control lists (ACLs) to specify which ArcSight Console users have permissions to view, run or edit different commands.

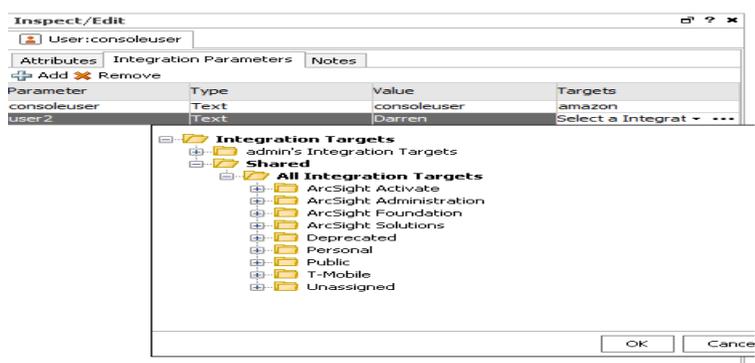
The following topics explain how to set up login details and ACLs in integration commands.

Setting User Login Parameters

You can specify login credentials on user accounts or on remote target servers. It is best to set login credentials on user accounts, but both options are described below. (Login credentials are not required for Connector integration commands because the authentication is handled as part of the SmartConnector setup.)

Setting Login Credentials

For URL commands on remote targets and script commands that run locally, you can define login credentials as a part of user configurations. (Choose **Navigator > Users**, select and edit a user or create a new one, then click the **Integration Parameters** tab on the **User Editor**.)



Defining login information as part of user accounts gives you the flexibility to configure multiple users, each with different logins. In this case, login credentials are not tied to the command target, but rather associated with individual users.

A single user account can have login credentials for different servers and scripts. In the example pictured above, the user “Darren” has login credentials for an appliance and also for a Logger appliance (which takes a user name and password as authentication).



Tip: Security best practices

- Always set login credentials (passwords or authentication tokens) to type “Password” (not “Text”). Credentials set to “Text” are not masked on the UI.
- Save authentication information only as parameters on user accounts, not on target servers. This strategy binds authentication details to specific users, and gives you tracking information based on user logins (for example, you can tell which users ran which commands and when).

Examples of authentication information are user name and password combinations, and authentication tokens sent in URLs.

Setting Login Credentials on Target Servers

Although not generally recommended, login credentials for URL commands on remote targets also can be defined as part of the Target definition, as described in [Specifying Targets](#). Choose **Navigator > Integration Commands > Targets** tab, select and edit a target or create a new one, then click the **Integration Parameters** tab on the Targets Editor.



If login information is defined here, everyone who uses the command uses the same credentials to log in to the remote target server.



Caution:

- Do not save authentication information as parameters on target servers. It runs the risk of opening up a remote server to any user who has access to the integration commands. Additionally, it does not give you any tracking information based on user logins to the server.
- Always set login credentials (passwords or authentication tokens) to type “Password” (not “Text”). Credentials set to “Text” are not masked on the UI.

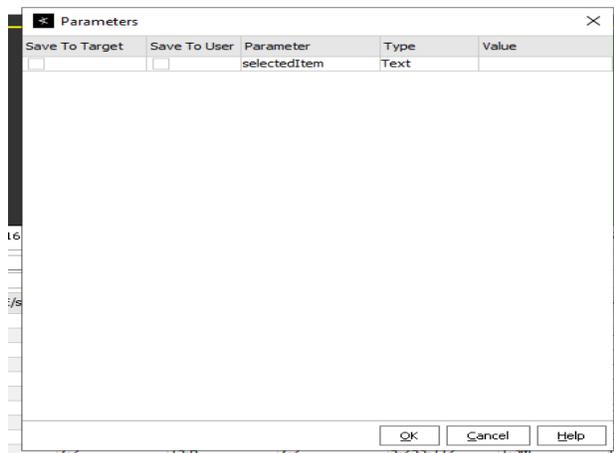
Setting Logins and Other Parameters to Prompt for Values at Runtime

You can set parameters for which you would like to prompt users to specify values at runtime, such as user name and password, host names, IP addresses, and other options.

When an integration command runs (that is, when you select an integration command in some context on the ArcSight Console), the command first looks for any required parameter values in a variety of sources, including in the command statement itself, in the defined context, on the user account, on the target (if there is one), and so forth. If it does not find parameter values in any of these places, the system prompts you to enter the values.

You can include login and other parameters as flags on a script command that runs against a server, as shown here for the archive command which runs on a Manager. When this command is run, it prompts the user for a Manager host name and administrator password. (It does not prompt for the user name, admin, since this already is provided in the command statement.)

```
archive -action import -m $hostmgr -u admin -p $passwd -f abc.arb
```



For an example of the run-time prompts returned by this command, see [Entering/Saving Command Parameters at Runtime](#).

Access Control Lists (ACLs) on Integration Commands

You can configure access control lists (ACLs) on integration commands, since they are resources (and ACLs can be configured on resources). For details on how this works in general, please see the information about granting or removing resource permissions in the [ArcSight Console User's Guide](#).

You can grant or limit read/write access to integration commands, integration configurations, and integration targets down to the *grouped resource* level for particular *user groups* by setting the setting ACL permissions on the resource group for any set of commands. Note that both the resources themselves and the users must both be in groups in order to work with them in this way.

For example, suppose you have a group of connector commands. The commands are grouped in a command group called “Investigate Commands,” and associated with a configuration called “Investigate Configurations.” You have two users (Darren and Larry) in a group called “Analyzers” to whom you want to give permissions to simply *run* these commands (not edit them). To do this, you would choose **Users** in the Navigator, select the Analyzers group, right-click and choose **Edit Access Control**. On the Resources tab, add both the Investigate commands and configurations groups, and give read access on both. (Add the resource Integration Command and select the appropriate command group in the selector, and add the resource Integration Configuration and select the appropriate configuration group in the selector, then click the Read check boxes for each under Resource Targets and save the ACLs for the user group.)



Tip:

- User group ACLs with **read** permissions on the integration command and configuration resources groups can **run** commands.
- User groups with **read and write** permissions on integration command and configuration resource groups can **run and edit** these commands.
- Commands can be configured to prompt for input parameter values when the command runs. If you want to give users permissions to **save the parameter values** required at command runtime, then you also must give **read and write** permissions to the associated **Integration Targets** groups on the ACL editor for the user group.

You can organize users and the commands, configurations, and targets into various groups to fit with the permissions schemes you want to create. You might, for instance, want to create one set of Investigate commands/configurations and give those permissions to one set of users (for example, Darren and Larry in the Analyzers group). Then you could create another set of Quarantine commands/configurations and give those permissions to a different group of users (for example, Samantha, Endora, and Arthur in Analyzer Administrators group). It might be appropriate for this second group to have more authority, and therefore you would grant a broader set of permissions to it (for example, both Investigate and Quarantine permissions per the ACL settings on the Analyzer Administrators group).

Running Integration Commands

After commands are configured, they are available in various contexts in the ArcSight Console.

For example, suppose you have a configuration for a set of commands with the contexts set as follows:

Location	Type	Selection	Data Type
Viewer	All Views	All Selections	IP Address

This means that the given commands are available on right-click context menus on any view (for example, active channels, list views, chart views, dashboards, and so on). The user can select any row, cell, or area on a chart. In this context, only IP addresses can be provided as valid parameters to the command.

If one of the commands in this configuration was a **Quarantine Node** command, then to use the command you would do the following:

1. Bring up an active channel, session list, active list, dashboard, or other resource in the viewer that shows, for example, a suspicious device, machine, or user that you want to quarantine.

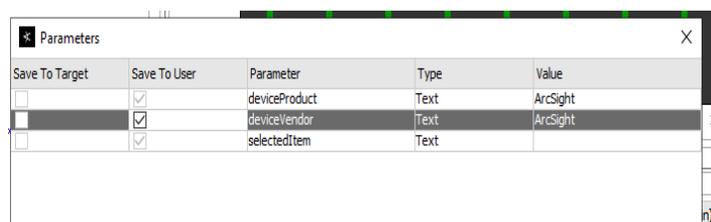
2. Find the row on the Viewer display that contains the suspicious entity, and select a cell in that row that contains the source IP address (for example, Attacker Address).
3. Right-click over the cell with the source IP address (for example, Attacker Address), and choose **Integration Commands > Quarantine Node**.

This launches the selected command, using the IP address for the selected cell as the parameter for the command.

In general, a right-click any context in the ArcSight Console UI for which integration commands have been configured show all integration configurations.

Entering/Saving Command Parameters at Runtime

Commands can be configured to prompt for parameter values at runtime (as described in [Setting Logins and Other Parameters to Prompt for Values at Runtime](#)). Also, if ready-made commands (such as for Logger) are not pre-configured, you are prompted for values. For example, parameters might ask for a particular host name as command input, an IP address against which to run a command, or login credentials to a target server.



If you launch a command that prompts for input, enter the appropriate text in the “Value” field for each required parameter.

If you have appropriate permissions, you have the option to save parameter values with the target or with your user account so that you don’t have to re-type them each time you run the command.



Tip: Security Best Practices

- To save parameter values at runtime, you need to belong to a group with read and write permissions to the associated targets.
- Always set login credentials (passwords or authentication tokens) to type “Password” (not “Text”). Credentials set to “Text” are not masked on the UI.
- Always save login credentials to user (Save to User), not to the target server. This strategy binds authentication details to specific users. This better safeguards access to the remote server to appropriate users. Also, you have tracking information based on logins as to which users entered which commands.

If you save authentication details as parameters on a target, you run the risk of opening up a remote server to any user who has access to the integration commands (but not necessarily an account on the target server). And you have no per-user tracking information.

Examples of authentication information are user name and password combinations, and authentication tokens that are sent in URLs.

Creating New Configurations On-the-Fly

You can also create a new integration configuration from within a context. To do this, right-click anywhere in the UI, and choose **Integration Commands > New Configuration**. See [Using Configurations to Group Commands](#) for next steps.

Network Tools as Integration Commands

The following standard network tools are also provided as integration commands. You can find this toolset in: /Integration Commands/Shared/ArcSight System/Tools/. You can edit these or add new commands, configurations, and contexts as described in [Defining Commands](#) and [Using Configurations to Group Commands](#).

With network tools integration commands you can:

- **Define contexts for where tools show up on the ArcSight Console.** You can customize integrated network tools and configure them for all types of views (charts, graphs, tables), and in the navigator, editors, and so on. Legacy network tools are available only on grid views; you cannot define the context.
- **Select and run commands on navigator tree items, all types of views, and editors items.** With integrated network tools, you can select various items in chart and graph views, on the editors, and in the navigator tree. Legacy network tools are limited to running only on the selected cell in a grid view (table) in the Viewer.
- **Configure access control lists (ACLs).** You can grant or limit access to integrated network tools commands for particular user groups by setting the setting ACL permissions on the tools resource group. The integrated network tools reside under /All Integration

Configurations/ArcSight System/Tools. You can control access to the tools commands and configurations groups (select the Tools group, right-click, and choose **Edit Access Control**). For more information, see the information about granting or removing resource permissions in the [ArcSight Console User's Guide](#). You can organize users and the tools themselves into various groups to fit with the permissions scheme you want to create. With the legacy network tools, you do not have this ACL option. See [Access Control Lists \(ACLs\) on Integration Commands](#) for more information.

Tree	Icon	Resource
Nslookup		Resolves an IP address to a host or domain name or vice versa.
Ping		Determines whether a particular IP address is online and/or it tests and debugs a network by sending a packet and waiting for a response.
PortInfo		Lists standard usage such as WWW or FTP, for a specified port number.
Traceroute		Shows the path from the ArcSight Console to the IP address selected in the grid view, reporting the IP addresses of all routers in between.
WebSearch		Search the Web through Google to find links to the keywords present in currently selected active channel grid view cells.
Whois		Looks up who is behind a given domain name; information might include addresses and telephone numbers.

These are configured with default Velocity Expressions for parameters. You can edit the commands and configurations for these network tools as needed (and add new ones).

To run a network tool, select an IP address in a grid view (for example, active channel, list, data monitor) and select **Integration Commands > <Network Tool>** from the context menu (for example, **Integration Commands > ping**).



Note:

- The Send Logs command is not configured as an integrated command. For more information, see the information about using network tools and send logs in the [ArcSight Console User's Guide](#).
- To add or re-configure the legacy tools, choose **Tools > Local Commands > Configure**, select a tool, and click **Edit**. Keep in mind that they have limitations compared to the new tools.

Send Documentation Feedback

If you have comments about this document, you can [contact the documentation team](#) by email. If an email client is configured on this computer, click the link above and an email window opens with the following information in the subject line:

Feedback on Developer's Guide (SmartConnector)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to Documentation-Feedback@microfocus.com.

We appreciate your feedback!