



# Resilient Incident Response Platform

HP ARCSIGHT INTEGRATION GUIDE v24

---

© 2015 Resilient Systems, Inc. All rights reserved.

This guide and the software described in this guide are furnished under a license accompanying the software and may be used only in accordance with the terms of such license. By using this guide, you agree to the terms and conditions of that license.

Resilient and Resilient Systems are trademarks or registered trademarks of Resilient Systems, Inc. in the United States and other countries. All other product names mentioned herein may be trademarks or registered trademarks of their respective companies.

Published: October 2015

<https://www.resilientsystems.com>

## Table of Contents

<b>1. Overview .....</b>	<b>5</b>
<b>2. ArcSight Case and Event Escalation Utility .....</b>	<b>6</b>
2.1. Installation.....	6
2.2. Mapping Template .....	6
2.3. Configuration.....	6
2.4. Testing .....	7
<b>3. ArcSight Case Export Processor Utility .....</b>	<b>8</b>
3.1. Mapping Template .....	8
3.2. Usage .....	8
<b>4. Web URL Integration .....</b>	<b>9</b>
4.1. Creating the Integration Command .....	10
4.2. Creating the Configuration .....	12
4.3. Testing the Integration .....	15
4.4. Adding Threat Information to Existing Incidents .....	16
4.4.1. Creating the Integration Command .....	16
4.4.2. Create your Configuration .....	18
4.4.3. Testing the Integration .....	20
<b>5. Using Custom Fields .....</b>	<b>21</b>



# 1. Overview

This document describes the techniques you can use to integrate the Resilient Incident Response Platform with HP ArcSight to simplify and streamline the process of escalating and managing incidents.

The Resilient platform is an open system with many integration points, ranging from user-configurable connections, simple scripts, and complex custom workflows. The integration techniques for HP ArcSight include:

- Automated escalation. Resilient Systems provides two utilities for automated escalation from ArcSight rules. One utility is for ArcSight cases and is a simple, configurable utility that processes “export files” and creates corresponding incidents and related artifacts. The other utility is for ArcSight correlated events, base events and cases. It is a flexible and configurable Python package that creates or updates incidents when an ArcSight Execute Command action is triggered in a rule.
- “Right-click” or Web URL integration. You can extend the ArcSight user interface to add custom commands and configurations. In these configurations, a simple customization creates a direct link from the ArcSight user interface where users can immediately create and enrich incidents in the Resilient platform.
- Email. The Resilient Email Connector can be used to process emails from the ArcSight server, extract information from each email, and then populate an incident and related information in the Resilient platform. This provides automatic and easily managed integration with every user’s familiar tools. To use this technique, see the *Resilient Incident Response Platform Email Connector Installation and Configuration Guide*.
- Custom integration. The Resilient REST APIs can be used for extensive custom integration. Additionally, the Resilient Actions Module can trigger custom actions when incident status changes in the Resilient platform. The Resilient API includes documentation and examples of these patterns in several programming languages. The result is a deep and powerful combination of product features, producing a streamlined response workflow. For more information, see the *Resilient Incident Response Platform Action Module Programmer’s Guide*.

## 2. ArcSight Case and Event Escalation Utility

The ArcSight case and event escalation utility is designed to be used for ArcSight correlated events, base events and cases. The utility is a flexible and configurable Python package that creates or updates incidents when an ArcSight Execute Command action is triggered in a rule.

You run the script run from ArcSight rules. It creates a new Resilient incident when the rule is executed. It can also update any existing incident for this case or event. You can populate the incident details from any information in the case, correlated event, base events, or other information available in ArcSight when the event is fired.

### 2.1. Installation

The script requires Python version 2.7. This is later than the default Python version in Red Hat and CentOS, so if ArcSight Manager is running on those OSs, you may need to install Python version 2.7.

The Resilient REST API is accessed with a helper module, `co3`, which should be used for all Python client applications. The `co3` module is a part of the Resilient REST API utilities `co3-api`. Download and install that first, following its instructions.

This integration also requires a few other Python dependencies. To install the dependent modules, use the following command:

```
pip install -r requirements
```

### 2.2. Mapping Template

The mapping from ArcSight events to Resilient incidents is done using a mapping template file. This uses the Jinja2 (<http://jinja.pocoo.org/docs/dev/templates/>) markup scheme.

The default name of the template file is **case\_template.jinja** when mapping from ArcSight cases, or **event\_template.jinja** when mapping from events. You can specify your own template using the **--template** parameter, which allows you to have different behavior from different ArcSight rules.

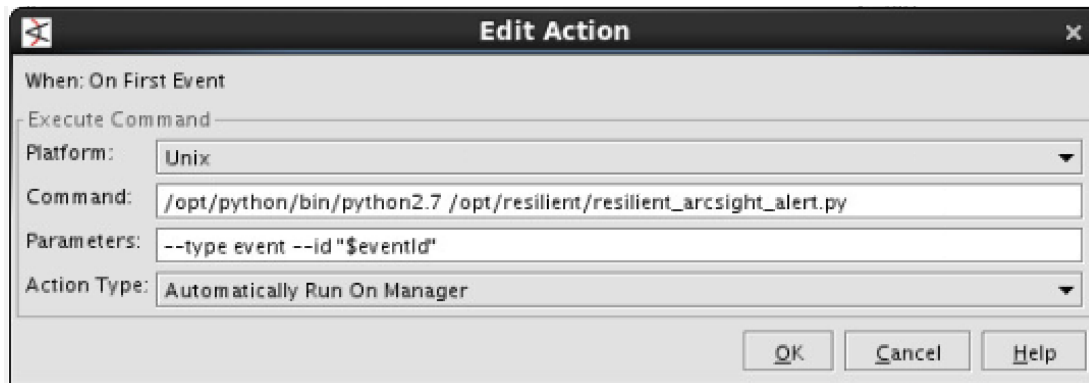
The supplied templates map a minimal set of ArcSight fields to the incident, and create a "simulation" incident (`"inc_training": true`). Modify them to your own needs before production use.

### 2.3. Configuration

General configuration parameters are set in **resilient\_arcsight\_alert.config**. Edit this file before you provide authentication credentials for the Resilient platform and ArcSight, and other required values. Make sure that this is readable only to the ``arcsight`` user.

In your Resilient platform, add a custom field, **arcsight\_id**, of type Text. This stores the event id or case id that triggered the incident, and allows incidents to be updated when rules are fired multiple times.

To run the script from a rule, add an **Execute Command** action, set to **automatically run on Manager**:



Set the Command to:

```
/path/to/python2.7 /path/to/resilient_arcsight_alert.py
```

To escalate from a case, set the Parameters to:

```
--type case --id $resourceid"
```

To escalate from a correlated event, set the Parameters to:

```
--type event --id $eventId"
```

Optionally you can add as many other parameters as you want; they are made available to the mapping template as **args[0]**, **args[1]**, etc.

**IMPORTANT:** Every time the script runs, it creates a new incident in the Resilient platform, which requires manual action by the incident response team. Do not trigger this script from a large volume of events.

## 2.4. Testing

To test manually, set the **--dry-run** command-line option. This produces the output JSON to represent the new incident without actually creating incidents. You can then test the script directly from the command-line, specifying the ID of a correlated event from ArcSight (the "Event ID" from Event Details), or the resource ID of an ArcSight case.

```
python resilient_arcsight_alert.py --type event --id <event_id> --dry-run
```

The output shows the structure of the ArcSight event then the structure of the Resilient incident produced by mapping through the template. This information is also written to the log.

## 3. ArcSight Case Export Processor Utility

The ArcSight Case Export Processor utility is used for ArcSight cases. The utility processes “export files” and creates corresponding incidents and related artifacts. You can configure ArcSight to export correlated events and cases automatically, resulting in automated escalation.

The utility reads case files that are exported by ArcSight into the manager's archive/export directory. These case files are XML documents that contain all the details of the case. The utility maps their contents to JSON that then creates an incident in the Resilient platform mapped from the case's contents.

### 3.1. Mapping Template

The mapping from ArcSight cases to Resilient incidents is done using a mapping template file. This uses the FreeMarker (<http://freemarker.org/docs/dgui.html>) markup scheme. An example template file is supplied in **arcsight.json.ftl**. You can use this as-is, or make a copy and edit to your own needs.

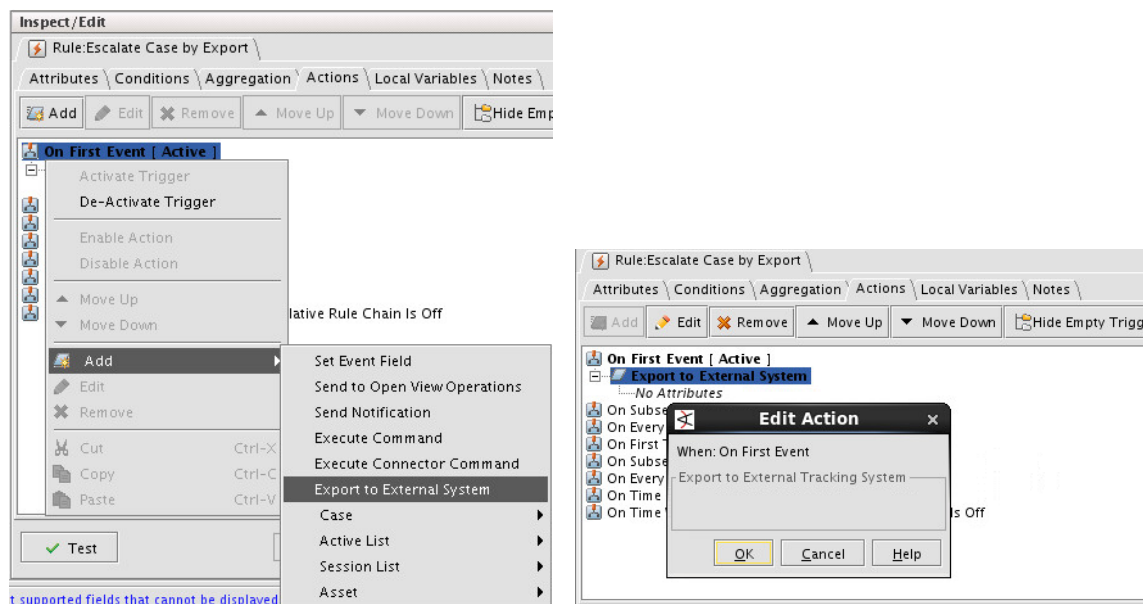
### 3.2. Usage

You run the utility from the command line. It can run once and process a single export file and stop, or it can watch a directory for new files and process them as they arrive. To see the available command-line options, run:

```
incident-builder -h
```

Normally, you run with the -w and -d options so that the utility continually watches the export directory, processes files that arrive, and moves them into another directory once they have been processed.

To generate the XML export files for processing, you must configure the ArcSight rules to export a case. The following screenshots show an example of how to do this.

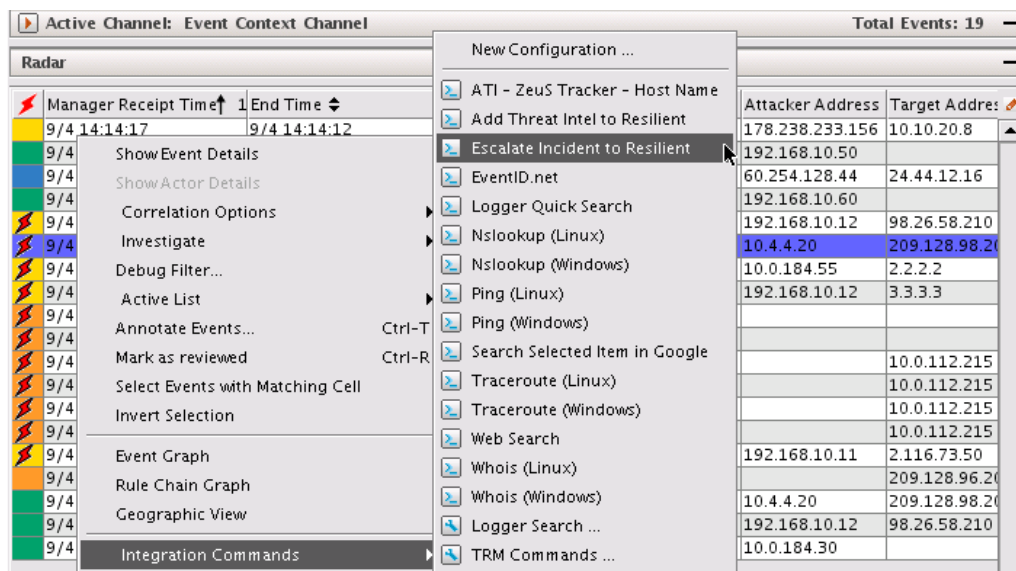




## 4. Web URL Integration

By directing a user's browser to specially-constructed web URLs, the user can be guided through an automatic creation of an incident, and other functions. A typical use case for this integration is in manually creating Resilient incidents from within ArcSight. This streamlines the process of escalation to the incident response team.

You can extend the ArcSight user interface to add custom commands and configurations. In these configurations, a simple customization creates a direct link from the ArcSight user interface where users can immediately create and enrich incidents in the Resilient platform.



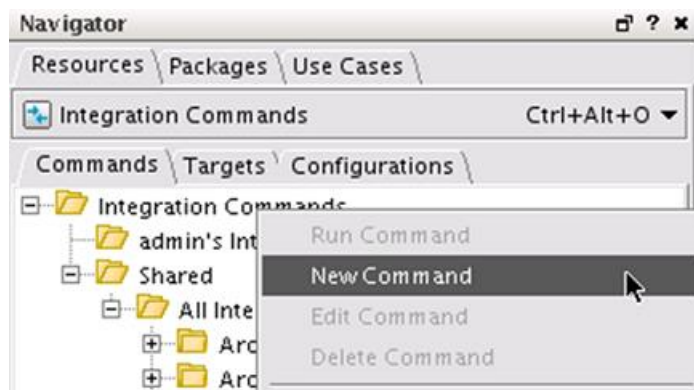
This integration uses the ArcSight **Integration Commands** feature. The following sections describe how to configure a manual escalation facility, which creates a new incident in the Resilient platform starting from the ArcSight user interface. There are three basic steps: create an integration command, create the configuration to carry out the command then test the integration.

For more information about Web URL integration, see the *Resilient Incident Response Platform Web URL Integration Guide*.

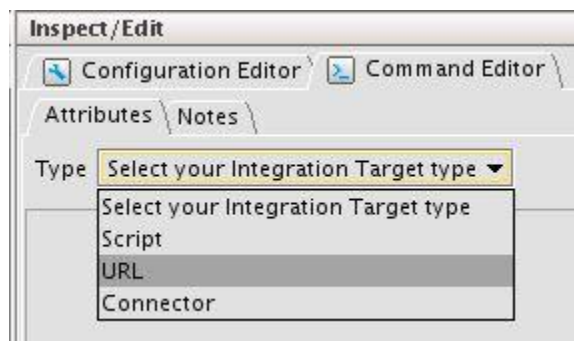
## 4.1. Creating the Integration Command

Perform the following to create the integration command:

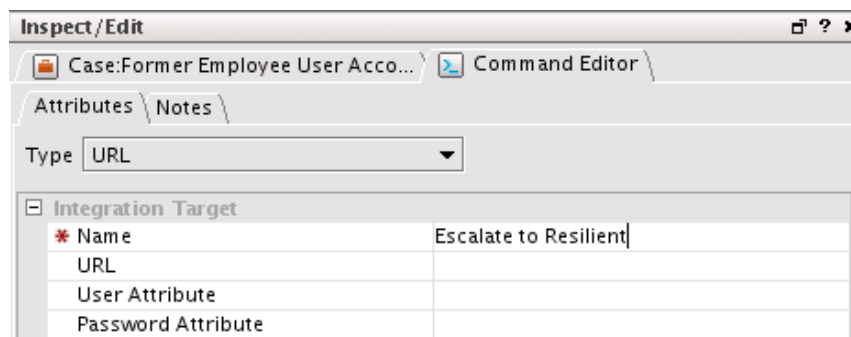
1. In the drop down menu on the left side of the ArcSight console, choose **Integration Commands**.
2. Make sure you have the Commands tab selected then right click on a folder and select **New Group** to create a new group folder called **Resilient**.
3. Right click your new folder and choose **New Command**. The command editor opens.



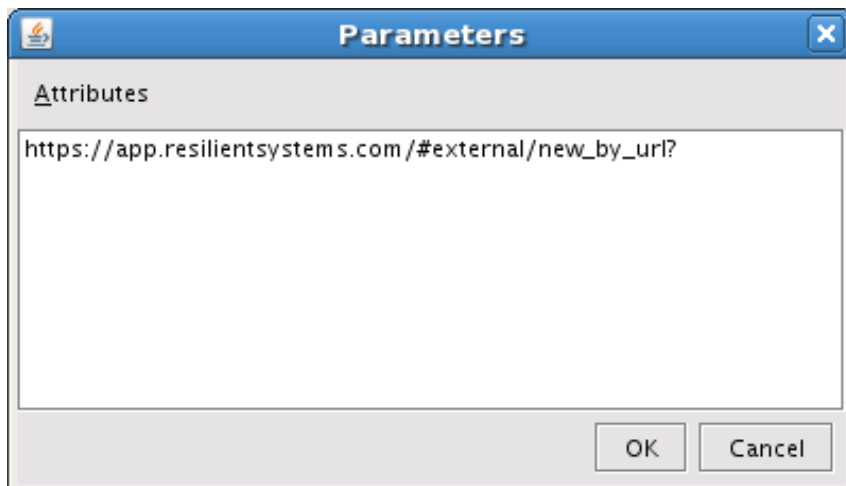
4. In the command editor box, select **URL** as the Integration Target Type.



5. Give your command a name, such as **Escalate to Resilient**.



6. In the URL field, click on the ellipsis (...) button then enter the URL where the information is to be sent. The parameters dialog appears.



7. In this box, enter the base URL and configure the fields that you wish to map to your Resilient incident.

The base URL is your normal Resilient platform URL, followed by **#external/new\_by\_url?**. For example:

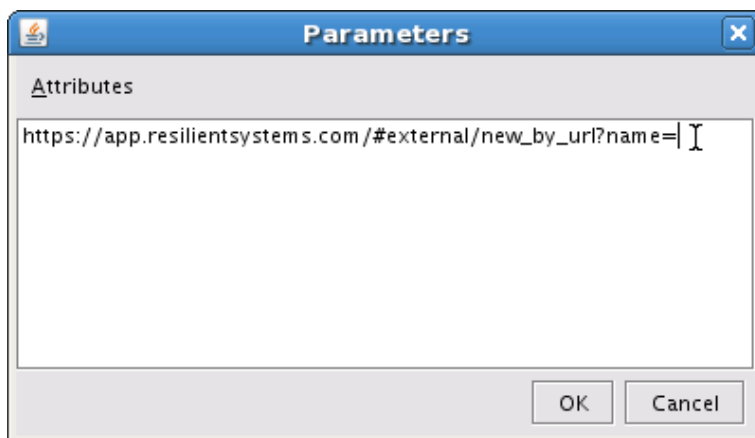
```
https://app.resilientsystems.com/#external/new_by_url?
```

8. After the '?', specify the parameters that map ArcSight field values onto fields in the Resilient incident. You can create static parameters that never change, as well as variable parameters that carry values from ArcSight into the Resilient platform. Separate multiple parameters by the ampersand (&).

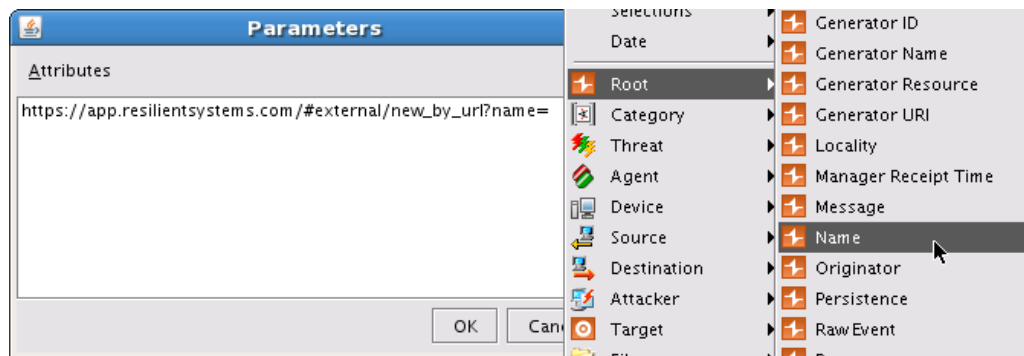
For example, the following is a full URL with parameters to set the name of the new incident to the word **New**, and a description of **Incident**.

```
https://app.resilientsystems.com/#external/new_by_url?&name=New&description=Incident
```

9. Use the ArcSight Parameters dialog to map the ArcSight values to the URL parameters. This is done by choosing a desired Resilient parameter and adding it to the base URL. You then right-click in the parameters box and choose the ArcSight field. Each parameter appears as a name like **\${address}**.



10. Right click to add the ArcSight field Root->Name.



11. Optionally, you can add more fields to your URL. You can also add multiple ArcSight field values into the value of a single URL parameter. For example, you could add an attacker address to your description, as well as the target. The URL would then look like this:

```
https://app.resilientsystems.com/#external/new_by_url?name=${name}&description=The attacker address was ${attackerAddress} and the target address was ${targetAddress}
```

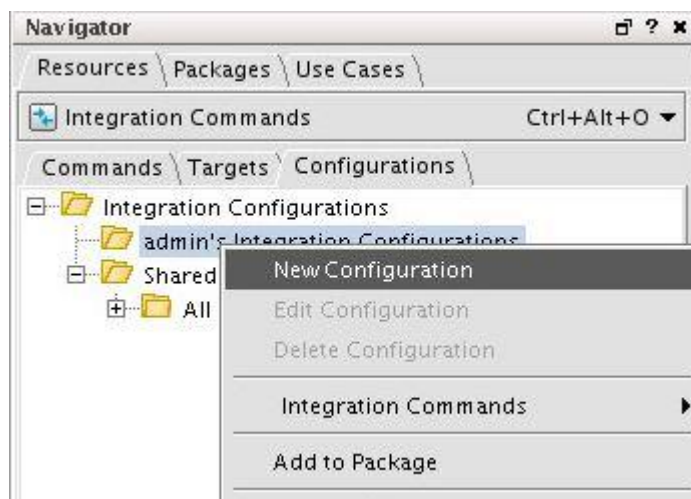
12. Once your URL is configured, click **OK** to save your work then click **OK** again to save your command in the command editor dialog box.

A Resilient incident has an extensive set of fields that can be configured in this way. Refer to the *Resilient Incident Response Platform Web URL Integration Guide* for a list of the standard Resilient field names, valid values for the standard set of coded fields, such as country, and additional techniques to specify fields of different types.

## 4.2. Creating the Configuration

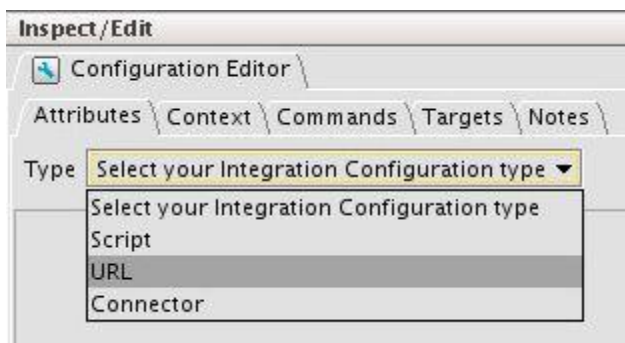
The configuration options determine how the command operates, such as where it is available in your ArcSight console. To create a configuration, perform the following:

1. Make sure that you are still in the Integration Configuration option in the dropdown list, as shown on the right side of the screen.
2. Select the **Configurations** tab.
3. Right click your folder and choose **New Group** then name your group **Resilient**.
4. Right click your new Resilient folder and choose **New Configuration**.

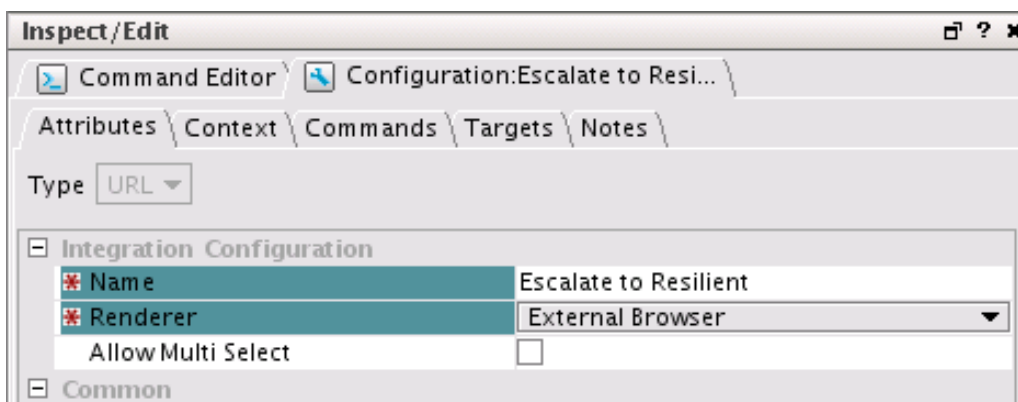


This opens the Configuration Editor on the right side of the screen.

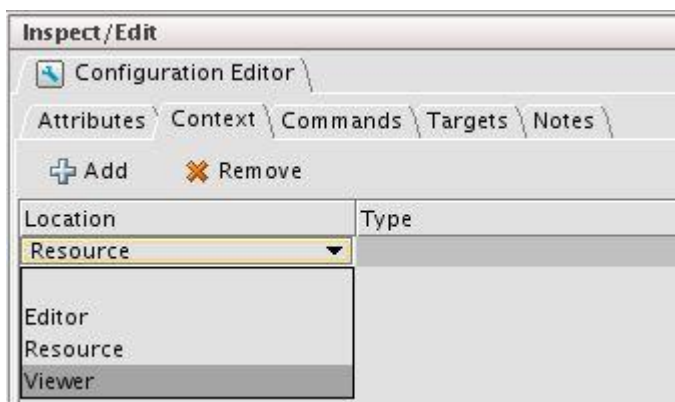
5. Within the Configuration editor on the Attributes tab, locate the dropdown menu that allows you to select your Configuration type. Choose **URL** from the list.



6. Give your configuration a name and choose how the URL is launched. Name your configuration, e.g., **Escalate to Resilient**, and choose **External Browser** for the renderer.

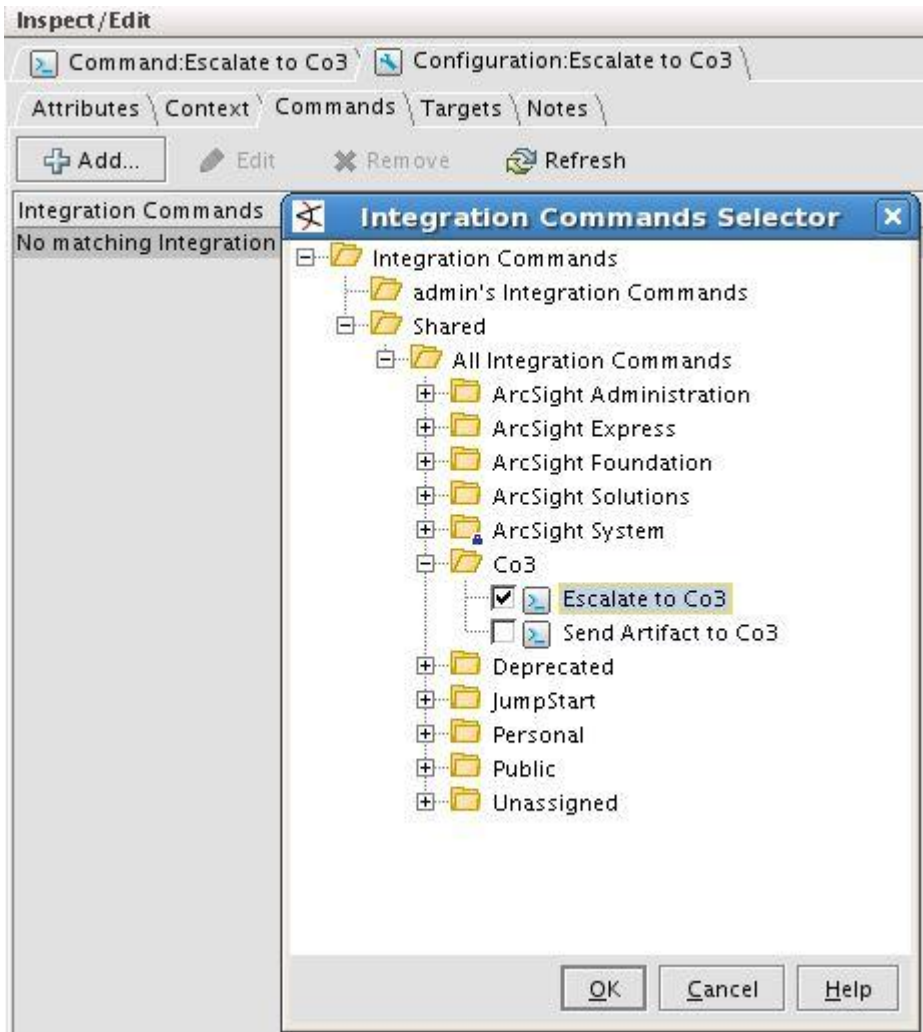


7. Click on the **Context** tab and then click on the **Add** button.
8. Click in the cell underneath **Location** and choose **Viewer** from the dropdown list.



9. Click on the **Commands** tab, and then click on the **Add** button. A new box appears.

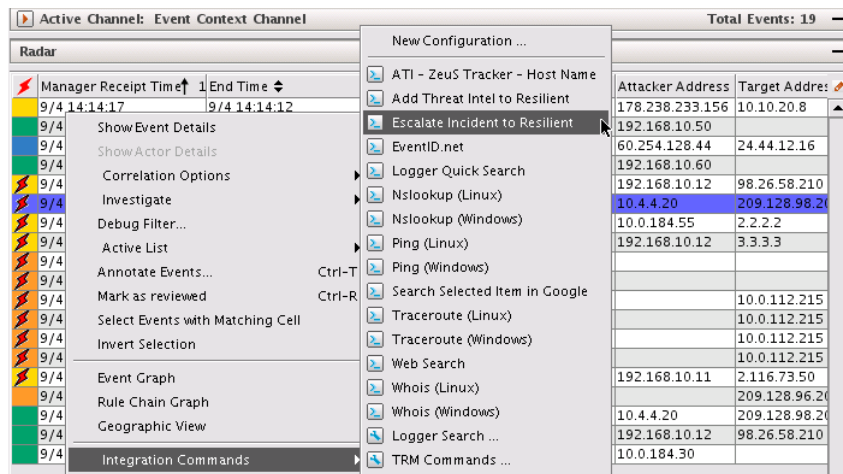
- Find your Resilient folder and the Integration Command that you configured previously, then click the checkbox next to it to select this command into the configuration.



- Click **OK** to save your selection, then click **OK** again at the bottom of the Configuration editor to save your work.

## 4.3. Testing the Integration

You can test your integration command by right clicking an event in the ArcSight Viewer panel, navigating to Integration Commands, and choosing **Escalate to Resilient**.



This opens an external browser providing a login prompt for the Resilient platform, if you are not already logged in. Once logged in, you see the details of the incident being created including the values being mapped from the ArcSight fields. From here, you can approve or change your selections and click the **Create** button to create your new incident in the Resilient platform.

[Dashboards](#)
[List Incidents](#)
[New Incident](#)

### Create New Incident

Name \*

Incident 2015-07-01

Description

Marketing Web Server Breached

Harm Foreseeable

Unknown

Severity

High

Criminal Activity

Yes

NIST Attack Vectors

Web

Data Format

—

Incident Type

Malware

System Intrusion

Create

## 4.4. Adding Threat Information to Existing Incidents

Follow the steps below to configure enrichment, which adds threat information from ArcSight to an existing incident in the Resilient platform starting from the ArcSight user interface.

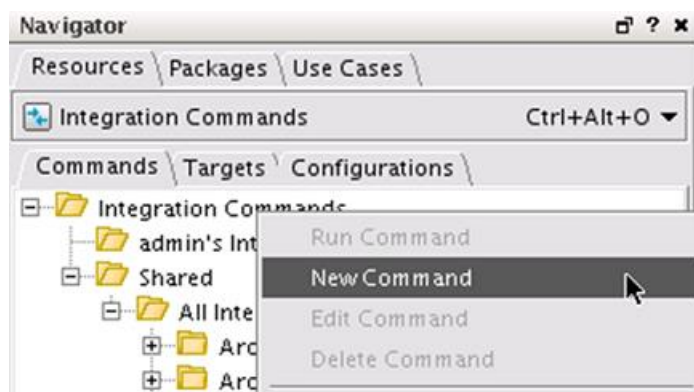
This process is very similar to the manual escalation steps previously detailed. The main differences include:

- The incident must already exist.
- The threat information is stored as Artifacts in the incident, and each artifact has a fixed and structured set of fields.

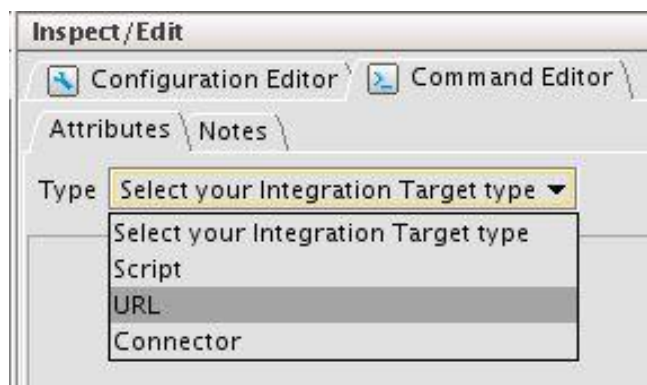
### 4.4.1. Creating the Integration Command

Perform the following to create the configuration command:

1. In the drop down menu on the left side of the ArcSight console, choose the **Integration Commands** option.
2. Make sure you have the Commands tab selected.
3. Select the folder, which can be the folder you already created (Resilient).
4. Right click the folder and choose **New Command**. The command editor opens.



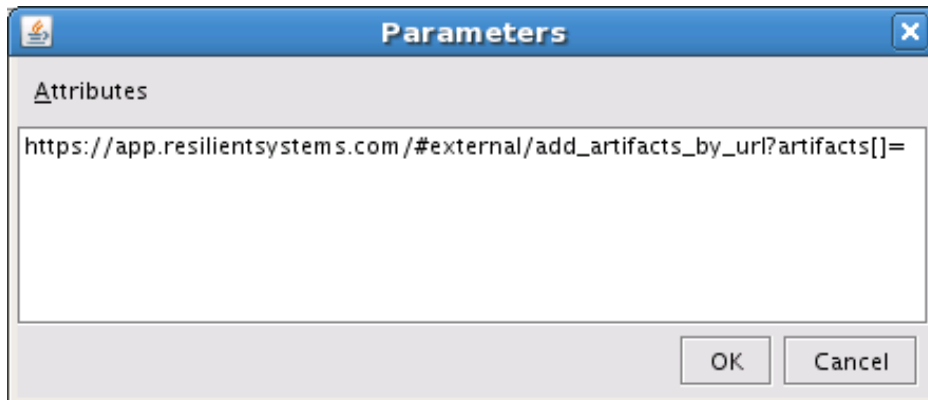
5. In the command editor box, select **URL** as the Integration Target Type.



6. Give your command a name, such as **Add Threat Intel to Resilient**.
7. In the URL field, click on the ellipsis (...) button then enter the URL where the information is to be sent. The Parameters dialog appears.



8. In this dialog box, enter the base URL and configure the fields you wish to map to artifacts in your Resilient Incident.



The base URL is your normal Resilient application URL, followed by “#external/add\_artifacts\_by\_url?artifacts[]=”. For example, enter the full URL below:

```
https://app.resilientssystems.com/#external/add_artifacts_by_url?artifacts[]=
```

9. After this base URL, configure a list of artifacts. Specify each artifact with a number signifying the artifact type, and then the artifact itself separated by a comma. For example, if the number 1 is used to indicate an IP address, and the IP address is 192.168.1.2, the resulting URL would be:

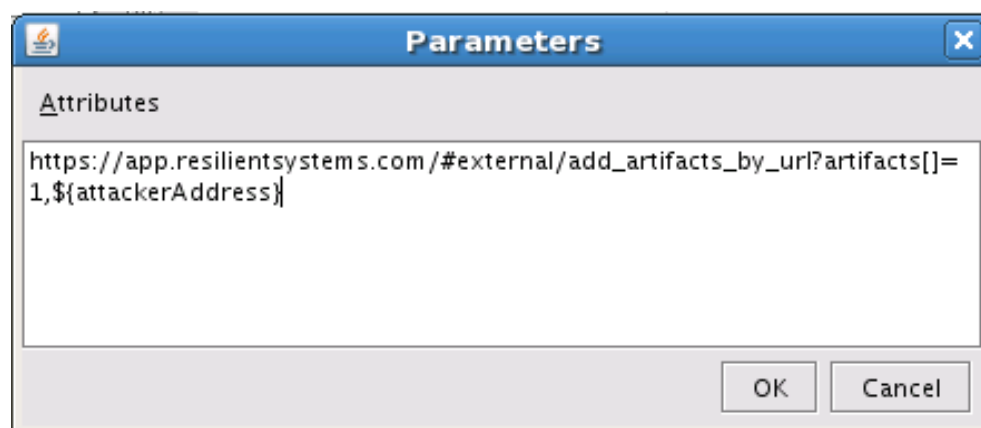
```
https://app.resilientssystems.com/#external/add_artifacts_by_url?artifacts[]=1,192.168.1.2
```

10. Optionally, you can add many more artifacts to the list in your URL, separated with commas.

Typical artifact types include IP addresses, DNS names, URLs. You can also add email subject, body, attachment name and sender. Refer to the *Resilient Incident Response Platform Web URL Integration Guide* for a list of the standard Resilient field names, valid values for the standard set of coded fields, such as country, and additional techniques to specify fields of different types.

11. For each artifact, you must map fields from ArcSight to the Resilient platform. This is done by choosing a Resilient artifact type ID, and adding it to the base URL. Afterwards, right-click the parameters box and select the ArcSight field to map. Each parameter appears as a name like **\${address}**.

The following example adds the Attacker Address.

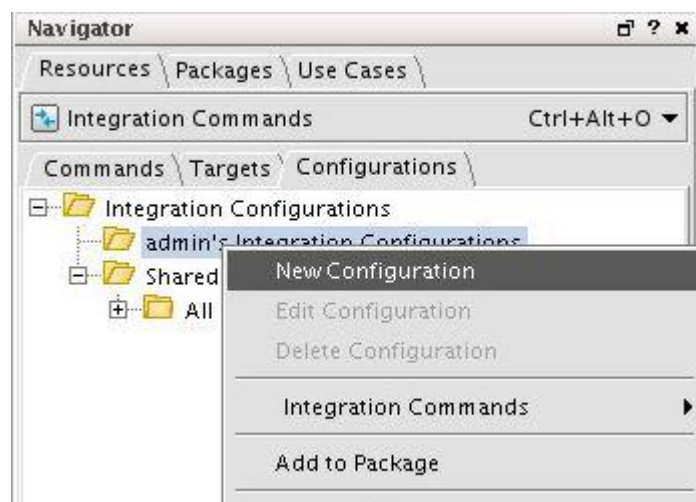


12. Once you configure your URL, click **OK** to save your work then click **OK** again to save your command in the command editor dialog box.

## 4.4.2. Create your Configuration

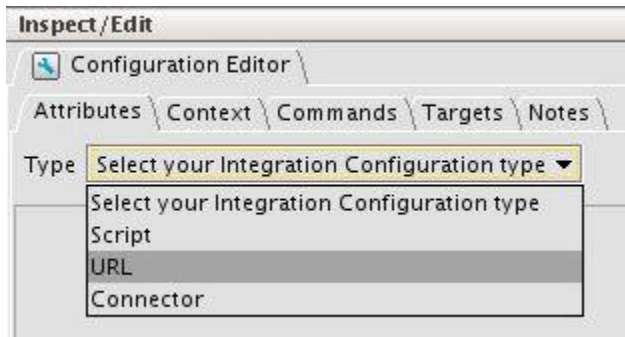
The Configuration options determine how the command operates, such as where it is available in your ArcSight console.

1. On the right side of the screen, make sure that you are still be on the Integration Configuration option in the dropdown list. Select the **Configurations** tab.
2. Right click your folder and choose **New Group** if necessary then name your group **Resilient**.
3. Right click your Resilient folder and choose **New Configuration**.

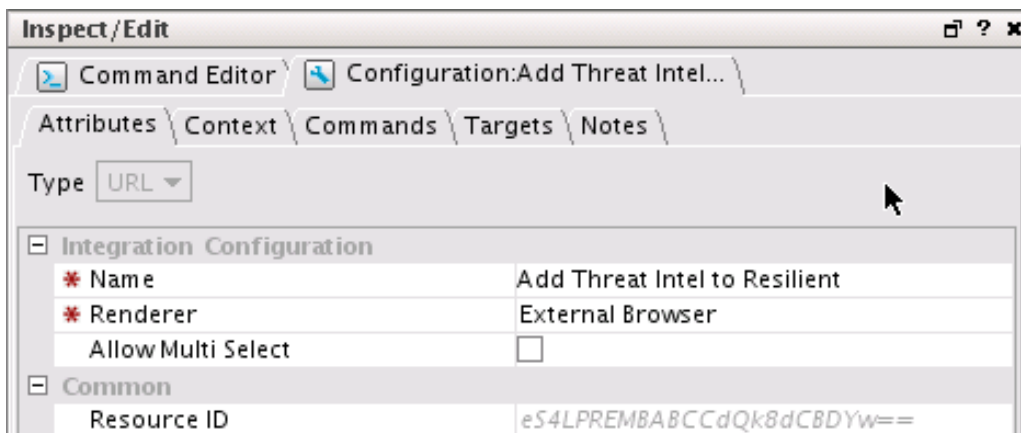


This opens the Configuration Editor on the right side of the screen.

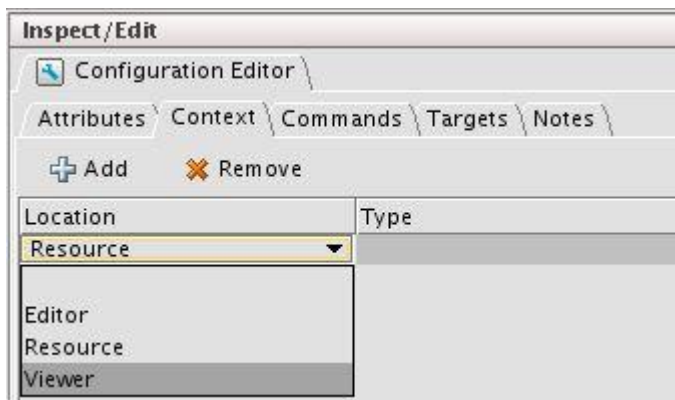
4. Within the Configuration editor on the Attributes tab, locate the dropdown menu that allows you to select your Configuration type. Choose **URL** from the list.



5. Give your Configuration a name and choose how the URL is launched. Name your configuration, e.g., **Add Threat Intel to Resilient**, and chose **External Browser** for the renderer.

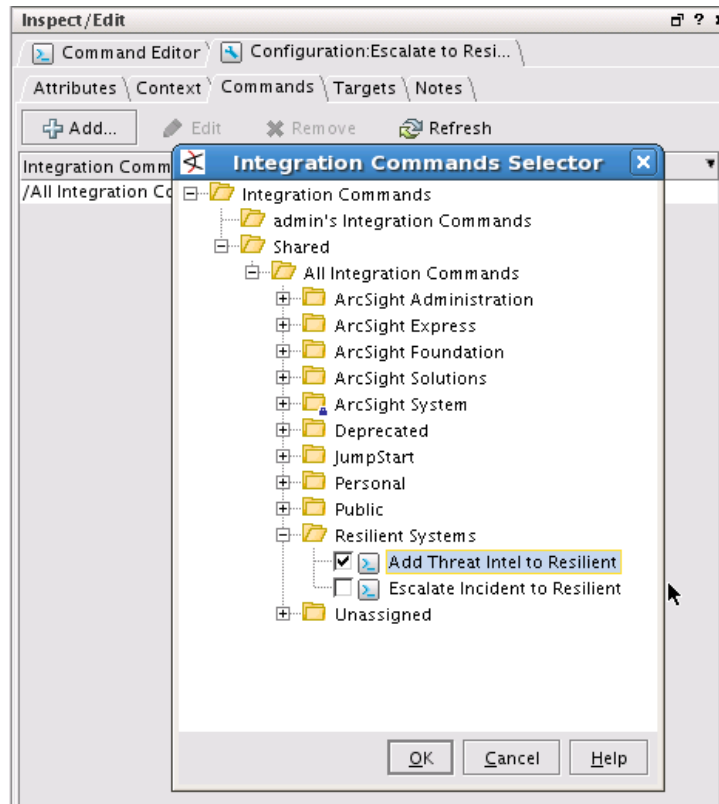


6. Click on the **Context** tab and then click on the **Add** button. Click in the cell underneath **Location** and choose **Viewer** from the dropdown list.



7. Click on the **Commands** tab, and then click on the **Add** button. A new box appears.

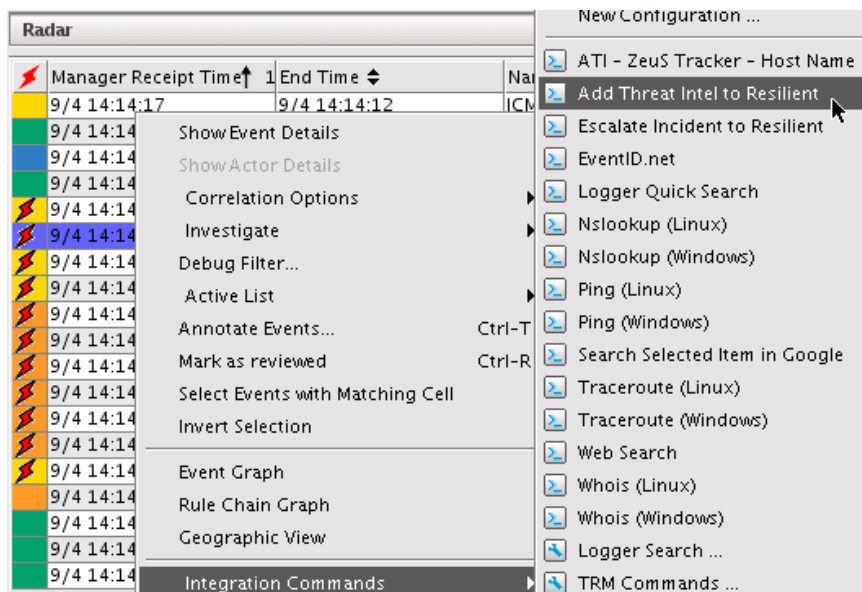
- Find your Resilient folder and the Integration Command that you configured previously, then click the checkbox next to it to select this command into the configuration.



- Click **OK** to save your selection, then click **OK** again at the bottom of the Configuration editor to save your work.

### 4.4.3. Testing the Integration

You can now test your integration command by right-clicking an event in the ArcSight Viewer panel, navigating to Integration Commands, and choosing **Add Threat Intel to Resilient**.



This opens an external browser providing a login prompt. Once logged in, you can add your Artifact by selecting the appropriate Incident from the dropdown list and clicking on the **Add Artifacts** button.

## 5. Using Custom Fields

You can further extend the ArcSight integration by the use of custom fields in the Resilient platform. Custom fields could potentially be created to consume the contents of any fieldset data in ArcSight.

To create a custom field in the Resilient platform, fill in the fields as follows:

1. Field name: Name that appears in the interface next to your custom field.
2. API Name: An automatically generated name for this field in the database for use with the API. You can modify this field if necessary.
3. Field type: Type of custom field. Options include:
  - a. Data picker: To select a date
  - b. Text: Single line text box
  - c. Number: Integer only field
  - d. Text area: Free form text input
  - e. Select: Dropdown with user-defined menu and options to define the type of scrolling, blank options, and default values

- f.** Boolean: Dropdown with selections for Yes, No, and Unknown
    - g.** Multiple Select: Like Select but with options to multi-select answers
  - 4.** Field operations: Used for the **Notifications** engine. Here you decide what operations would cause a notification to trigger, such as your custom field being selected. In such an instance, the **equals** option would be selected.
  - 5.** Field requirements: You decide if selecting your field is optional, required, or required upon close.
  - 6.** Field Tooltip: Decide if your field needs a pop up Tooltip and its contents.
  - 7.** Field placeholder: Example text that appears inside a custom field box.
- To address a custom field via the URL, use its **API name** in the integration.