User Interface Integration

ScienceLogic version 8.1.2
# Table of Contents

- **Introduction** .................................................................................................................. 4
  - What's In This Manual? ..................................................................................................... 4
- **Navigation Tabs** ............................................................................................................... 5
  - Overview .......................................................................................................................... 5
  - Viewing the Navigation Tab Editor Page ......................................................................... 5
  - Creating a Top-Level Tab .................................................................................................. 6
  - Creating an Entity-Level Tab ......................................................................................... 9
  - Editing and Deleting a Tab ............................................................................................. 12
- **Navigation Bars** ............................................................................................................. 14
  - Overview .......................................................................................................................... 14
  - Viewing the Navigation Bar Editor Entries .................................................................. 14
  - Filtering the List of Entries in the Navigation Bar Registry ........................................ 16
  - Creating a Navigation Bar Entry ................................................................................... 17
    - Creating a New Navigation Bar Entry From an Existing Entry .................................. 19
  - Editing a Navigation Bar Entry .................................................................................... 19
  - Deleting a Navigation Bar Entry .................................................................................. 20
- **Base Widgets** .................................................................................................................. 21
  - Overview .......................................................................................................................... 21
  - Common Fields ............................................................................................................... 21
  - Snapshot/Single Series > Performance > (base) Leaderboard/Top-N ............................ 22
    - Configuring the Leaderboard/Top-N Widget ............................................................... 22
    - Viewing the Leaderboard/Top-N Widget ................................................................... 28
  - Custom > Tools > (base) Context Iframe Content ......................................................... 33
    - Configuring the Context Iframe Content Widget ....................................................... 34
    - Viewing the Context Iframe Content Widget ............................................................. 37
- **Embedding Content from the ScienceLogic Platform in Another Web Page** ............... 38
  - Embedding a Page from the ScienceLogic Platform in Another Web Page ................ 38
  - Embedding a Single Widget from the ScienceLogic Platform in Another Web Page .... 39
- **What is a Proxied Web Service?** .................................................................................... 41
  - Overview .......................................................................................................................... 41
  - Defining a Proxied Web Service ...................................................................................... 42
  - Authentication Types ....................................................................................................... 42
  - XSL Transformations ....................................................................................................... 42
  - DNS-Based Proxy ............................................................................................................ 43
- **Creating a Credential for a Proxied Web Service** ......................................................... 45
  - Overview .......................................................................................................................... 45
  - Creating a Credential for HTTP Authentication ............................................................ 46
  - Creating a Credential for Custom Header Authentication ........................................... 47
  - Creating a Credential for PHP Script Authentication ................................................... 49
- **Defining Proxied Web Services** ..................................................................................... 54
  - Overview .......................................................................................................................... 54
  - Creating a Proxied Web Service ...................................................................................... 54
  - Viewing the List of Proxied Web Services ..................................................................... 56
  - Editing and Deleting a Proxied Web Service ................................................................ 57
  - Deleting a Proxied Web Service ..................................................................................... 58
- **Using the Proxied Web Service Widget** ....................................................................... 59
  - Overview .......................................................................................................................... 59
  - The Proxied Web Service Widget .................................................................................. 59
- **XSL Transformations** ..................................................................................................... 62
  - Overview .......................................................................................................................... 62
Uploading Service Files .................................................................................. 63
Viewing The List of Service Files .................................................................. 64
Editing a Service File .................................................................................... 64
Creating a Proxy XSL Transformation .............................................................. 65
Viewing the List of Proxy XSL Transformations .............................................. 66
Editing and Deleting a Proxy XSL Transformation ....................................... 67
Deleting a Proxy XSL Transformation ........................................................... 67
Associating an XSL Transformation with a Proxied Web Service .................. 68
Using an XSL Transformation with the Proxied Web Service Widget ............. 68

DNS-Based Proxy .......................................................................................... 71
What is a DNS-Based Proxy? ........................................................................... 71
Viewing a List of DNS-Based Proxies .............................................................. 72
Prerequisites for Defining a DNS-Based Proxy .............................................. 73
Defining a DNS-Based Proxy .......................................................................... 73
Editing a DNS-Based Proxy ........................................................................... 75
Deleting a DNS-Based Proxy ......................................................................... 75

Example Using a Custom Header for Authentication ...................................... 77
Overview ......................................................................................................... 77
Configuring Splunk to Accept Requests from the ScienceLogic platform ........ 78
Testing the Request ....................................................................................... 79
Creating a Credential .................................................................................... 80
Defining a DNS-based Proxy .......................................................................... 81
Defining a Proxy Web Service in the ScienceLogic platform ......................... 83
Define a Dashboard Widget to Display the Results .......................................... 84

Example Using HTTP Authentication .............................................................. 86
Overview ......................................................................................................... 86
Testing the Request ....................................................................................... 86
Creating an XSL Transformation .................................................................... 88
Creating a Credential .................................................................................... 89
Defining an XSL Transformation in the ScienceLogic Platform ..................... 91
Defining a Proxy Web Service in the ScienceLogic Platform ......................... 93
Defining a Dashboard Widget ......................................................................... 94

Example Using a PHP Script for Authentication ............................................ 96
Overview ......................................................................................................... 96
Creating the Credential ................................................................................ 97
Defining a Proxy Web Service and a PHP Script in the ScienceLogic platform . 98
Defining a Dashboard Widget to Display the Results ....................................... 100
What's In This Manual?

This manual describes how to configure the user interface of the ScienceLogic platform to display information from external systems. You can do this with:

- Custom Navigation Bars
- Custom Navigation Tabs
- Top-N and Context iFrame widgets
- Proxied Web Services

Additionally, this manual describes how to embed pages from the user interface of the ScienceLogic platform in an external application. Each of these methods is described in this manual.
Chapter 2

Navigation Tabs

Overview

The ScienceLogic platform allows you to create custom top-level tabs, or add a tab in the Device Administration panel, the Organization Administration panel, or the Asset panel. The new tab can lead to an external page in your network (for example, a web-based control panel), to another page in the platform, or to an external website.

Top-level tabs appear with the other default tabs in the platform (for example, the [Views] tab or the [Tickets] tab).

Entity tabs can be created for the Device Administration panel, Organization Administration panel, and/or the Asset panel. You can define access permissions for top-level and entity tabs, and define where they are visible.

Viewing the Navigation Tab Editor Page

The Navigation Tab Editor page displays a list of tabs that have been created in the ScienceLogic platform. From the Navigation Tab Editor page you can also create, edit, and delete a top-level or entity tab. To view a list of created tabs in the Navigation Tab Editor page:

1. Go to the Navigation Tab Editor page (System > Customize > Navigation Tabs).
2. The Tab Registry pane at the bottom of the Navigation Tab Editor page displays a list of created tabs. For each tab, the Navigation Tab Editor page displays the following information:
- **Element Type.** Specifies if the tab is a top-level tab or entity level tab.
  - A top level tab is defined as *Main Admin UI*.
  - An entity level tab is displayed as *Device, Organization, or Asset*.

- **ID.** Numeric identifier automatically assigned to the tab by the ScienceLogic platform.

- **Tab Title.** Name given to the tab.

- **Access.** Type of user account that can access the tab. Options are *Administrators or Users*.

- **User Edit.** User who created or last edited the tab.

- **Date Edit.** Time the tab was created or last edited.

## Creating a Top-Level Tab

In the **Navigation Tab Editor** page you can create a top-level tab that will appear with the default top-level tabs in the ScienceLogic platform. The new tab can contain only a URL and will display the page specified in the URL you have designated. The URL can lead to a page in the platform or a page in an external website (for example, a web-based control panel).

To create a new top-level tab:

1. Go to the **Navigation Tab Editor** page (System > Customize > Navigation Tabs).
2. In the **Navigation Tab Editor** page, select the [Reset] button to clear any fields from the Editor pane.
3. In the Editor pane at the top of the page, supply a value in each field:
• **Navigation Tab Location.** Specifies whether to display the new tab as a top-level tab in the ScienceLogic platform or as part of the **Organization Administration** panel, **Device Administration** panel, or **Asset** panel. Select:
  ○ *Main Admin UI.* The new tab will appear as a top-level tab in the platform.

• **Navigation Tab Title.** Tab's label. This is the text that users will see on the tab.

• **Link Type.** For future use. Do not use this field.

• **Always Visible.** Specifies whether or not users who are not allowed to access the tab will be able to view the tab in the platform. Choices are:
  ○ *No.* Tab will not appear in the platform for users who do not have the appropriate permission keys to access the tab.
  ○ *Yes.* Tab will always appear in the platform and will be visible to users who do not have the appropriate permission keys to access the tab.

• **Display For.** If you selected *Main Admin UI* in the **Navigation Tab Location** field, this field is grayed out.

• **Entity Selector.** If you selected *Main Admin UI* in the **Navigation Tab Location** field, this field is grayed out.

• **Access.** Users who will be allowed to access the tab, based on the type of user account. The choices are:
  ○ *Administrators.* Only users with account type "administrator" are allowed to access this tab.
  ○ *Users.* Both users with account type "user" and users with account type "administrator" are allowed to access this tab.

• **Permission Keys.** Select zero, one, or more access keys in this field. To access this tab, a user account must be granted at least one of the selected access keys. Access Keys are defined by the ScienceLogic system administrator from the **Access Keys** page (System > Manage > Access Keys).
- **Target.** Specifies how the URL will be opened after the tab has been selected. Choices are:
  - _self. The specified URL will open and replace the ScienceLogic platform in your browser.
  - _blank. The specified URL will open in a new tab or new window, and the ScienceLogic platform will remain available in a separate tab or window.
  - Zoombox. The specific URL will be opened in a new window in front of the ScienceLogic platform.
  - iframe. The specific URL will open in the pane below the header and top-level navigation tabs. For example, if you select iframe and enter "http://www.sciencelogic.com/" in the URL field, the following page is displayed when the tab is selected:

![ScienceLogic Platform](image)

- **URL.** Full URL or link for the page that will be displayed in the tab.
4. Select the [Save] button to save the tab. The tab will be displayed with the default top-level tabs and will lead to the URL you have specified.

Creating an Entity-Level Tab

An entity-level tab can be created in the Device Administration panel, Organization panel, or the Asset panel. You can create an entity-level tab from the Navigation Tab Editor page. To create an entity-level tab:

1. Go to the Navigation Tab Editor page (System > Customize > Navigation Tabs).
2. In the Navigation Tab Editor page, select the [Reset] button to clear any fields from the Editor pane.
3. In the Editor pane at the top of the page, supply a value in each field:
Navigation Tab Location. Specifies whether to display the new tab as a top-level tab in the ScienceLogic platform or as part of the Organization Administration panel, Device Administration panel, or Asset panel. Select:

- **Entity Page(s).** The new tab will appear as part of the Organization Administration panel, Device Administration panel, or Asset panel.

Navigation Tab Title. Tab's label. This is the text that users will see on the tab.

**Link Type.** For future use. Do not use this field.

Always Visible. Specifies whether or not users who are not allowed to access the tab will be able to view the tab in the platform. Choices are:

- **No.** Tab will not appear in the product for users who do not have the appropriate permission keys to access the tab.
- **Yes.** Tab will always appear in the product and will be visible to users who do not have the appropriate permission keys to access the tab.

Display For. In this field, you specify the area in the ScienceLogic platform where you want the tab to appear. The choices are:

- **Organization.** The new tab will appear in the Organization Administration panel for selected organizations.
- **Device.** The new tab will appear in the Device Administration panel for selected devices.
- **Asset.** The new tab will appear in the Asset Administration panel for selected assets.

Entity Selector. Depending on the entity you selected in the Display For field, you can select which organization/device/asset(s) will display the new tab in this field. You can choose to display the new tab in the panel for all, one, or multiple organizations/devices/assets.

- **To select all entities, select the entry All Organizations/Devices/Assets**
- **To select a single organization/device/asset, highlight it.**
- **To select multiple organizations/devices/assets, left click while holding down the <Shift> key.**

Access. Users who will be allowed to access the tab, based on the type of user account. The choices are:

- **Administrators.** Only users with account type "administrator" are allowed to access this tab.
- **Users.** Both users with account type "user" and users with account type "administrator" are allowed to access this tab.

Permission Keys. Select zero, one, or more access keys in this field. To access this tab, a user account must be granted at least one of the selected access keys. Access Keys are defined by the system administrator from the Access Keys page (System > Manage > Access Keys). To learn more about access keys and permissions, see the Access Permissions manual.
- **Target.** Specifies how the URL will be opened once the tab has been selected. Choices are:
  - **_self.** The specified URL will open and replace the ScienceLogic platform in your browser.
  - **_blank.** The specified URL will open in a new tab or new window, and the ScienceLogic platform will remain available in a separate tab or window.
  - **zoombox.** The specific URL will be opened in a new window in front of the ScienceLogic platform.
  - **iframe.** This option is available only if Device is selected in the **Display For** field. The specific URL will open in the pane below the read-only information in the **Device Administration** panel. For example, if you select **iframe** and enter "http://www.sciencelogic.com/" in the **URL** field, the following page is displayed when the tab is selected:

  ![Image](image.png)

- **URL/Link.** Full URL or link for the page that will be displayed in the tab.
Select the [Save] button to save the tab. The tab will be displayed in the **Device Administration** panel, **Organization Administration** panel, or **Asset Administration** panel as you have specified. For example, if you created an entity-level tab for the **Device Administration** panel, it could appear similar to the image below:

![Device Administration Tab](image_url)

---

**Editing and Deleting a Tab**

After you have created a custom navigation tab you can edit and delete it from the **Navigation Tab Editor** page. You can edit one or more parameters of a custom tab. You can also delete a custom tab.

To edit a custom tab:

1. Go to the **Navigation Tab Editor** page (System > Customize > Navigation Tabs).
2. In the **Navigation Tab Editor** page, select the [Reset] button to clear any fields from the Editor pane.
3. In the **Tab Registry** pane at the bottom of the page, find the tab you want to edit. Select its wrench icon (🔧).
4. The Editor pane (at the bottom of the page) is populated with values from the selected tab.
5. To save your changes, select the [Save] button.
6. If you would like to save your changes as a new tab without making changes to the tab you selected to edit, supply a new name in the **Navigation Tab Title** field and select the [Save As] button. The new tab will be added to the **Navigation Tab Editor** page.

You can delete a custom navigation tab from the **Navigation Tab Editor** page. You can delete only one custom tab at a time.
To delete a custom navigation tab:

1. Go to the **Navigation Tab Editor** page (System > Customize > Navigation Tabs).

2. In the **Tab Registry** pane, find the navigation tab you want to delete. Select its bomb icon (💥).

3. You will be prompted to confirm the tab deletion. Select the [OK] button to delete the navigation tab. Select the [Cancel] button to return to the **Navigation Tab Editor** page.
Overview

The ScienceLogic platform allows you to add new entries to the NavBar panes that appear to the left of some pages. You can add new entries to the NavBar panes for the following top-level tabs:

- [System]
- [Registry]
- [Reports]
- [Preferences]
- [Knowledge]

NOTE: Advanced users can associate a custom-written Java executable with a navigation bar entry. Contact ScienceLogic for information on creating a new navigation bar entry using a Java executable.

Viewing the Navigation Bar Editor Entries

You can view the list of all existing entries in each navigation bar for the tabs listed above in the Navigation Bar Editor page. The Navigation Bar Editor page contains two panes:

- The Navigation Bar Editor, where you can create and edit custom Navigation Bar entries.
- The Navigation Bar Registry, where you can view a list of all Navigation Bar entries, including the default entries defined by ScienceLogic.
This section will discuss the **Navigation Bar Registry** pane. To view the list of all existing navigation bar entries:

1. Go to the Navigation Bar Editor page (System > Customize > Navigation Bars).
2. The **Navigation Bar Editor** in the lower half of the page displays the entries for each tab. By default, the registry pane displays a list of entries that appear under the [System] tab. You can view the entries for other tabs by using the **Interface Filter** drop-down. To learn how to use the **Interface Filter** drop-down, see the section *Filtering the List of Entries in the Navigation Bar Registry*.
3. The **Navigation Bar Registry** pane displays the following about each entry:

   ![Image of Navigation Bar Registry pane](image)

   - **Interface**. The top-level tab in the platform where the NavBar entry will appear. The choices are:
     - System
     - Registry
     - Reports
     - Preferences
     - Knowledge Base
   - **Sub-Menu**. Sub-heading within the navigation bar, under which the navigation bar entry will appear.
   - **Name**. Name of the navigation bar entry.
   - **Access**. Users who will be allowed to access the navigation bar entry, based on the type of user account. The choices are:
- **Administrators.** Only users with account type “administrator” are allowed to access this navigation bar entry.
- **Users.** Both users with account type “user” and users with account type “administrator” are allowed to access this navigation bar entry.

- **Pre-Order.** Defines the sort order for the sub-menu.
- **Sub-Order.** Sort order for the entries in each sub-menu. The lowest numbered entry appears first under the sub-menu; the highest numbered entry appears last in the sub-menu.
- **Edited By.** Name of the user who last edited this entry.
- **Last Edited.** Date and time entry was last edited.

**Filtering the List of Entries in the Navigation Bar Registry**

The Navigation Bar Editor page includes an [Interface Filter] option to view entries for only a specific tab: [System], [Registry], [Reports], [Preferences], or [Knowledge].

To filter the list of entries in the Navigation Bar Registry:

1. Go to the Navigation Bar Editor page (System > Customize > Navigation Bars).
2. In the Navigation Bar Registry pane, select the [Interface Filter] field to filter the list of entries by a specific tab. The choices are:

   - **[System].** Only NavBar entries that appear in the [System] tab will be displayed.
   - **[Registry].** Only NavBar entries that appear in the [Registry] tab will be displayed.
   - **[Reports].** Only NavBar entries that appear in the [Reports] tab will be displayed.
   - **[Preferences].** Only NavBar entries that appear in the [Preferences] tab will be displayed.
   - **[Knowledge].** Only Navigation Bar entries that appear in the [Knowledge] tab will be displayed.
After selecting one of the filter options, the entries in the **Navigation Bar Registry** pane will be filtered to display entries from only the selected tab.

### Creating a Navigation Bar Entry

You can create a custom navigation bar entry to link to an external website or page in the ScienceLogic platform. You can create a custom navigation bar entry from the **Navigation Bar Editor** page.

To create a new navigation bar entry:

1. Go to the **Navigation Bar Editor** page (System > Customize > Navigation Bars).
2. In the **Navigation Bar Editor** page, select the [Reset] button to clear any fields in the **Navigation Bar Editor** pane.
3. In the **Editor** pane, define values in the following fields:
   - **Sub-Menu.** Top-level category in the navigation bar under which the new entry will appear. This category is not a link itself, but instead is a heading under which to group links. For example, for the NavBar entry that leads to the current page, this value would be "Customize". You can specify an existing sub-menu or define a new sub-menu. This field can contain any combination of alphanumeric characters, with a maximum length of 32 characters.
   - **Name.** Name of the link as it will appear in the NavBar. This field can contain any combination of alphanumeric characters, with a maximum length of 32 characters.
   - **Interface.** The top-level tab in the ScienceLogic platform where the NavBar entry will appear. The choices are:
     - [System]
     - [Registry]
     - [Reports]
     - [Preferences]
     - [Knowledge]
   - **Target.** Specifies the HTML "target" for the new link. This field specifies how the URL will be displayed. The choices are:
     - *Embedded.* Includes a browser plug-in in the current window.
     - *Popup.* Opens the URL in a separate window, on top of the current page.
     - *iframe.* Opens the URL in a frame within the current page.
     - *_self.* Opens the URL in the currently active window (usually the frame to the right of the navigation bar).
     - *_blank.* Opens the URL in a new window.
- **URL/Link.** The URL of the page that is displayed when a user selects the navigation bar item. This can be the URL of a page in the platform or the URL of a page outside the platform. This field can contain any combination of alphanumerical characters, with a maximum length of 128 characters. Forward slash (/), underscore (_), and question mark (?) are allowed.

- **Access.** Users who will be allowed to access the NavBar entry, based on the type of user account. The choices are:
  - Administrators. Only users with account type "administrator" are allowed to access this navigation bar entry.
  - Users. Both users with account type 'user' and users with account type "administrator" are allowed to access this NavBar entry.

- **Parent Item.** This field is not currently used.

- **Note/Comment.** Any notes or comments about the entry.

- **Pre-Order.** Sort order for the sub-menu. The lowest numbered sub-menu appears first in the navigation bar; the highest numbered sub-menu appears last in the navigation bar. If all sub-menus in a navigation bar have the same number, the platform sorts the sub-menus alphabetically.

  **Note:** When creating a new entry in a navigation bar entry and in an existing sub-menu, you must retain the pre-order already defined for the sub-menu.

- **Sub-Order.** Sort order for the entries in each sub-menu. The lowest numbered entry appears first under the sub-menu; the highest numbered entry appears last in the sub-menu. If all entries in a sub-menu have the same number, the platform sorts the entries alphabetically.

- **Access Key(s).** The access key required to access the entry in the NavBar. Access Keys define the tabs and pages users have access to and the actions that a user may perform. These Access Keys are defined by the system administrator from the Access Keys page (System > Manage > Access Keys). To learn more about Access Keys, see the Access Permissions manual.

4. Select the [Save] button to save the new entry. To view the new entry, go to the page where the new entry resides and refresh the page. The new entry will appear in the NavBar.
Creating a New Navigation Bar Entry From an Existing Entry

You cannot edit a default NavBar entry. However, you can create a new navigation bar entry from an existing default entry. For example, if you wanted to create a new navigation bar in the "Customize" sub-menu, you could add an entry by selecting an existing NavBar and editing its information. To create a new NavBar entry from an existing entry:

1. Go to the Navigation Bar Editor page (System > Customize > Navigation Bars).
2. Use the Interface Filter field to find the existing entry you want to copy. In the Navigation Bar Registry pane (at the bottom of the page), find the entry you want to use as a template and select its grayed-out wrench icon.
3. The Editor pane at the top of the Navigation Bar Editor pane will be populated with the values from the selected entry. You can edit one or more of these fields.
4. To save the new navigation bar entry, select the [Save As] button. The information for the original, existing NavBar entry you selected will not be overwritten.
5. To view the new entry, go to the page where the new entry resides and refresh the page. The new entry will appear in the NavBar.

Editing a Navigation Bar Entry

In the Navigation Bar Editor page, you can edit a custom entry for a NavBar. You cannot edit a default entry for a navigation bar.

To edit a custom entry for a navigation bar:

1. Go to the Navigation Bar Editor page (System > Customize > Navigation Bars).
2. In the Navigation Bar Editor page, use the Interface Filter field to find the entry you want to edit.
3. In the Navigation Bar Registry pane, find the entry you want to edit. Select its wrench icon.
4. The Navigation Bar Editor pane will be populated with values from the selected entry.
5. To save your edits, select the [Save] button.
6. To create a new NavBar entry without overwriting entry you selected, click the [Save As] button.
Deleting a Navigation Bar Entry

In the Navigation Bar Editor page, you can delete a custom entry from a NavBar. You cannot delete a default entry from a NavBar.

To delete a custom entry from a NavBar:

**CAUTION:** Use caution when deleting an entry from a NavBar. If you decide to delete an entry, be sure to note its field values, so you can restore the entry if necessary.

1. Go to the Navigation Bar Editor page (System > Customize > Navigation Bars).
2. In the Navigation Bar Editor page, use the Interface Filter field to find the entry you want to delete.
3. In the Navigation Bar Registry pane, find the entry you want to delete. Select its bomb icon (💣).

4. You will be prompted to confirm the NavBar entry. Select the [OK] button to delete the entry from the NavBar. Select the [Cancel] button to cancel the deletion and return to the Navigation Bar Editor page.
Overview

The ScienceLogic platform includes several built-in (base) widget definitions that allow you to access and display data from outside the ScienceLogic platform. This chapter describes how to configure these base widgets and what is displayed in each base widget. For descriptions of additional widgets, see the Dashboards manual.

Common Fields

The following fields appear in all widget configuration panes:

- **Widget Name.** Enter a title for the widget. This title is displayed in the header that appears at the top of the widget. If you leave the default value of "{auto}" in this field, the ScienceLogic platform will automatically generate a title for the widget based on what is currently being displayed in the widget.

- **Widget Refresh Rate.** Specify how frequently the widget will be automatically updated with new data. The choices are:
  - Widget Default. The widget will refresh at its default refresh rate, as defined by the widget developer. You can view and edit the default refresh rate in the Dashboard Widgets page (System > Customize > Dashboard Widgets) by selecting the wrench icon (🔧) for a widget.
  - Auto-refresh disabled. The widget will not automatically refresh.
  - 1 minute. The widget will automatically refresh every minute.
  - 5 minutes. The widget will automatically refresh every 5 minutes.
  - 10 minutes. The widget will automatically refresh every 10 minutes.
  - 15 minutes. The widget will automatically refresh every 15 minutes.
  - 30 minutes. The widget will automatically refresh every 30 minutes.
45 minutes. The widget will automatically refresh every 45 minutes.

1 hour. The widget will automatically refresh once an hour.

**NOTE:** For widgets with a *Display Type* option, if you do not select a display type for the widget, the dashboard automatically defaults to the first display type in the list of display types for that widget.

**Snapshot/Single Series > Performance > (base) Leaderboard/Top-N**

The *Leaderboard/Top-N* widget displays utilization statistics for a specific performance metric. The widget displays utilization for the devices with the highest or lowest values for the performance metric.

You can configure this widget to use a *Custom URL* that points outside the platform. The *Custom URL* field affects what is displayed when a user clicks on a data point in a graph. You can specify that an external URL is displayed. You can use variables to include information from the platform about the data point (like device name or interface ID) in the URL.

**Configuring the Leaderboard/Top-N Widget**

To configure a *Leaderboard/Top-N* widget, supply values in the following fields:
- **Leaderboard Config**. Select the devices or IT Services that will be included in the graph.
  - Specify whether the graph will display devices with the highest (Top) or lowest (Bottom) utilization for the selected metric.
  - Select the number of devices to include in the widget.
  - Select whether the Average, Minimum, Maximum, Total, or Standard Deviation of the metric over the selected time period should be used. Select the time period over which the metric should be evaluated.
  - **Use Timespan Context**. If you select this checkbox, the time period over which the metric will be evaluated can be selected in another widget in the dashboard.
  - **Ignore Min of 0**. If you select this checkbox, the widget will not include devices with a value of zero for the selected metric over the selected time period.

- **Collection Configuration**. Select the metric for the widget using the following fields:
  - **Collection Type**. Select the source of data that will be displayed in the widget.
  - **Collection**. Select the type of metric that will be displayed in the widget. The options available in this field are based on your selection in the **Collection Type** field.
- **Series.** If applicable, select the specific metric to display in the widget. The options available in this field are based on your selection in the **Collection** field.

- **Scale Prefix.** Allows the user to select a unit scale for the widget's y-axis that is appropriate for their data series. If the user does not select a scale prefix, the widget will auto-scale the y-axis to an appropriate scale based on the values being displayed.

- **Filter.** For Dynamic Applications, the Filter field will match indexes and/or labels; for interfaces the Filter field will match ifName/ifDescr/ifAlias or interface tags; for monitors (port, CV, process, etc.), the Filter field will match the appropriate unique element of that monitor (port number, hostname, etc.).

- **Device Filters.** Select which devices will be evaluated for inclusion in the widget. You can limit the devices that will be included in the widget by selecting one or more Organizations, Device Groups, IT Services, Device Categories, or Device Classes.

**NOTE:** In widgets that allow you to filter the list of devices by the device class or device category, merged devices include special behavior. For merged devices, you can select either the device class or device category of the physical device or the device class or device category of the component device. If both device classes or device categories are selected, a merged device will appear twice in a single widget.

- If you select the **Use Device-related Context,** the list of devices that will be evaluated for inclusion in the widget can be selected in another widget in the dashboard.

- **Misc.** Additional settings that affect the display of the graph.

  - **Use Old (Flash) Graphs.** This feature is no longer supported.

  - **Color by Device State.** If you select this checkbox, each graphical element in the dashboard will be colorized based on device state.

  - **Click/Link Behavior.** Select how the widget will behave if a user selects a graphical element in the widget.

    - **Auto-Select Device/Service.** When the dashboard is loaded the first entry in this widget is selected. The selected metrics control what is displayed in other widgets in the dashboard.

    - **No Action (Disabled).** No action is performed.

    - **Open Custom URL in Kiosk Mode.** Use the selected entity to populate the variable(s) in a custom URL (specified in the Custom URL field). Display the populated custom URL in a kiosk window. In the Custom URL field, you must specify a URL to populate.

    - **Open Custom URL in New Window.** Use the selected entity to populate the variable(s) in a custom URL (specified in the Custom URL field). Display the populated custom URL in a new window. In the Custom URL field, you must specify a URL to populate.

    - **Open in Kiosk Mode.** A new window opens and displays a time-series performance graph in kiosk mode. In kiosk mode, the options that are normally available in the **Device Performance** page will not be displayed. For example, if a service provider is configuring a dashboard for their customers, kiosk mode will allow the customer to drill-down to the raw data for a metric without giving them full access to the **Device Performance** page.
- **Open Performance Window.** A new window opens and displays a time-series performance graph of the selected metric.

- **Select Device/Service.** The selected metrics control what is displayed in other widgets in the dashboard.

  - Custom URL. Specify the custom URL that you want to populate with the Open Custom URL selection. You can include one or more variables in a custom URL. You can use the variables in place of a value from the ScienceLogic platform. Variables are surrounded in curly braces.

    You can click the wrench icon (🔧) in the Custom URL field to open the field in a larger window. This window includes a **Token Builder** that enables you to build variables into the custom URL. When you select a series of tokens in the **Token Builder** pane, the corresponding variables are inserted into the custom URL.

    The variables use the syntax:

    `{X.Y}`  

    where:

    - **X** is one of the following entities:
      - deviceObject. Contains attributes associated with the selected device.
      - interfaceObject. Contains attributes associated with the selected interface.
      - serviceObject. Contains attributes associated with the selected IT service.
      - timespan. Contains attributes associated with the selected timespan.

    - **Y** is an attribute for that entity. For all entities except timespan, the available attributes are the attributes from the API that do not return lists or links (i.e. single fields). For timespan, you can specify the following attributes:

      - `{timespan.start}`
      - `{timespan.end}`
      - `{timespan.duration}`

    - The following are the most commonly used device attributes:

      - **id.** The numeric ID of the device.
      - **hostname.** The hostname of the device discovered via hostname discovery.
      - **ip.** The IP address the ScienceLogic platform uses to communicate with the device.
      - **name.** The name of the device.
      - **organization.** The organization of the device. If you use this attribute, you must specify the organization attribute that you want to use. If you use this attribute, you must use the following variable syntax:
Any Base or Extended Custom Attributes that have been added to your ScienceLogic system.

The following are the most commonly used interface attributes:

- **device.** The device with which the interface is associated. If you use this attribute, you must specify the device attribute that you want to use. If you use this attribute, you must use the following variable syntax:
  
  ```java
  {context.interfaceObjects.<entity index>.device.<device attribute>}
  ```

- **organization.** The organization with which the interface is associated. If you use this attribute, you must specify the organization attribute that you want to use. If you use this attribute, you must use the following variable syntax:
  
  ```java
  {context.interfaceObjects.<entity index>.organization.<organization attribute>}
  ```

- **iFIndex.** The SNMP index associated with the interface.
- **ifDescr.** The description of the interface.
- **alias.** The alias of the interface.
- **name.** The name of the interface.

The following are the most commonly used IT service attributes:

- **service_id.** The numeric ID of the IT service.
- **service_name.** The name of the IT service.

The following are the most commonly used organization attributes:

- **company.** The name of the organization.
- **billing_id.** The billing ID of the organization.
- **crm_id.** The CRM ID of the organization.

For example, a custom URL could be:

```plaintext
http://my.website.com/{deviceObject.id}
```

where {deviceObject.id} is the device selected in another widget.

- **Legend Label Options.** Specifies the information that is included in the legend for the widget. Choices are:
  
  - **Element Names.** Displays the names of the device(s) or IT Service(s) in the legend.
  - **Index Label.** If you selected a performance Dynamic Application in the Collection field, you use the Index field to select the data series to display in the widget. If you select the Index Label checkbox, the widget includes the name of the index in the legend.

- **Title Label Options.** Each selected option will appear in the title of the widget. Choices are:
- **Window.** Displays the value from the **Widget Name** field.
- **Type.** Displays the value from the **Type** field.
- **Collection Type.** Displays the value from the **Collection Type** field.
- **Collection Name.** Displays the value from the **Collection** field.
- **Series Name.** Displays the value from the **Series** field.
- **Aggregation.** Displays the value from the **Aggregation** field.

- **Axis Label Options.** Select optional methods for labeling information on the X-axis. Choices are:
  - **Units.** If this check box is selected, numbers along the X-axis include units of measurement; otherwise, units of measurement are not included.

- **Display Type.** Select how the information will be displayed in the widget:
  - **Pie.** Widget will be displayed in a pie graph. Displays each event as percentage of total events. Slice color represents the severity of the event.
  - **Column.** Widget will be displayed in a bar graph. Displays number of occurrences on the y-axis. On the x-axis, displays each event in its own colored vertical bar. Bar color represents the severity of the event.
  - **Horizontal Bar.** Widget will be displayed in a horizontal bar graph. Displays number of occurrences on the x-axis. On the y-axis, displays each event in its own colored horizontal bar. Bar color represents the severity of the event.
  - **Radar.** Displays a multi-pointed, color-coded polygon or circle. Users determine if the radar is a polygon or circle by selecting Arc or Line in the **Grid Lines** field. Each point on the polygon or circle represents an event. The number of event instances is measured by the concentric rings. The number value of each concentric ring increases from center to perimeter.
  - **Spreadsheet.** Widget will be displayed as a spreadsheet. Displays each event in its own row. Each event has its own column, with number of occurrences. Clicking on the event name displays the **Event Console** page, with only the occurrences of the selected event displayed.
  - **Scoreboard.** For the selected events, displays each event name and the number of occurrences for each event. Events are ordered by severity, with critical first. Display is tally-style, in large format for easy viewing.
• **Display Options.** Specify how the graph will be formatted.
  
  - **Chart minimum.** You can accept the minimum value that the ScienceLogic platform determines, usually the lowest collected value or lowest calculated value (Automatic), or you can specify a minimum value (Fixed Value).
  
  - **Chart maximum.** You can accept the maximum value that the ScienceLogic platform determines, usually the highest collected value or highest calculated value (Automatic), or you can specify a maximum value (Fixed Value).
  
  - **Axis Scaling.** You can select either linear or logarithmic scaling for the widget.
  
  - **Threshold value.** You can select Enabled and then enter a threshold value in this field. The ScienceLogic platform will then include a heavy line in the graph that indicates the threshold.
  
  - **Severity ranges.** Select whether a low value (Increasing) or a high value (Decreasing) indicates a healthy state.
  
  - **Severity Slider.** Use the sliders and/or supply values in the slider fields to define the range of values at which the metric is in a healthy, normal, minor, major, and critical state. If you select Gauge in the **Display Type** field, you can also define the minimum value and maximum value that will be displayed in the gauge.

**Viewing the Leaderboard/Top-N Widget**

The **Leaderboard/Top-N** widget displays utilization statistics for a specific performance metric. The widget displays utilization for the devices with the highest or lowest values for the performance metric.

**NOTE:** If the **Leaderboard/Top-N** widget has been defined with the **Use Device-related Context** checkbox selected, and a selected Device Group or selected IT Service does not contain any devices, the **Leaderboard/Top-N** widget will display a message saying the context contains no devices.
For example, an instance of the **Leaderboard/Top-N** widget that is configured to display the 10 devices with the highest average latency in a bar graph looks like this:

![Image of Leaderboard/Top-N widget](image)

The **Leaderboard/Top-N** widget can be configured to display:

- Any performance metric collected by the ScienceLogic platform.
- The devices with the highest or lowest minimum, maximum, average, total, or standard deviation for the selected performance metric during the last frequent normalization period (5 - 30 minutes), hourly normalization period, or daily normalization period.
- 5, 10, 15, 20, 25, 30, 35, 40, 45, or 50 devices. The selection of devices that will be evaluated for highest or lowest utilization can be limited to only devices in specific organizations, device groups, device categories, or device classes.

The following table lists the required Access Hooks (in addition to the “Dash:View” and “Dash:View Shared” Access Hooks) that users need to view specific types of data in the **Leaderboard/Top-N** Widget:

<table>
<thead>
<tr>
<th>Data</th>
<th>Required Access Hook IDs</th>
<th>Required Access Hook Names</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Click Actions for Devices</td>
<td>DEV_PERF_REPORT_VIEW</td>
<td>Dev:Performance Graphs</td>
</tr>
<tr>
<td>Click Actions for IT Services</td>
<td>ITS_SERVICE_VIEW</td>
<td>IT Service: View</td>
</tr>
<tr>
<td>Content Verification</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>DLAG</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>DNS Policies</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>Dynamic Applications</td>
<td>DEV_VIEW SYS_DYN_APP_MANAGEMENT</td>
<td>Dev:View  System &gt; Manage &gt; Applications</td>
</tr>
</tbody>
</table>

Base Widgets
<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail Round Trip</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>File Systems</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Interfaces</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>IT Services</td>
<td>DGRP_VIEW</td>
<td>DevGroup:View</td>
</tr>
<tr>
<td>Port Monitors</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Process Monitors</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Transaction Verification</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Video Performance</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Vitals</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
<tr>
<td>Windows Service Monitors</td>
<td>DEV_VIEW</td>
<td>Dev:View</td>
</tr>
</tbody>
</table>

The **Leaderboard/Top-N** widget can be configured to display in one of the following formats:

- **Pie Chart**

![Pie Chart Diagram](image-url)
- **Column graph**

![Column graph example](image)

- **Horizontal Bar graph**

![Horizontal Bar graph example](image)
- **Radar chart**

![Radar chart image]

- **Spreadsheet**

![Spreadsheet image]
- **Scoreboard**

![Scoreboard Table]

Depending on the configuration of the widget, selecting a metric performs one of the following actions:

- The **Device Performance** page will open in a separate window with the graph for the selected metric displayed.
- Your selection will define what is displayed in other widgets in the dashboard. You can select multiple elements by holding down the `[Ctrl]` key (or `[Command]` on Apple computers).
- A custom URL will open in a separate window.
- No action will be performed.

## Custom > Tools > (base) Context Iframe Content

The Context Iframe Content widget accepts a URL as input and then displays the page specified by the URL. The URL can include device variables that are populated by the device context, (i.e. when you select a device in another widget.)
Configuring the Context Iframe Content Widget

To configure the Context Iframe Content widget, supply a value in the following field:

- **Context Iframe URL.** Enter the URL to display in the Iframe widget.
  - You can enter a relative or absolute URL.
  - You can use HTTP or HTTPS.
  - You can include variables that specify attributes of a device, interface, IT service, or organization selected in another widget.

Click the wrench icon (🔧) in the Context Iframe URL field to open the field in a larger window. This window includes a Token Builder that enables you to build variables into the URL. When you select a series of tokens in the Token Builder pane, the corresponding variables are inserted into the URL.

Variables in the Context Iframe URL field are specified in the following format:

```
{context.<entity type>.<entity index>.<attribute>}
```

The following are valid entity type values for variables:
- **deviceObjects.** Contains attributes associated with the devices selected in another widget.

- **interfaceObjects.** Contains attributes associated with the interfaces selected in another widget.

- **organizationObjects.** Contains attributes associated with the organizations selected in another widget.

- **serviceObjects.** Contains attributes associated with the organizations selected in another widget.

The following are valid values for specifying an entity index in variables:

- **first.** The attribute used to populate the variable will be from the first entity a user selected in another widget.

- **last.** The attribute used to populate the variable will be from the last entity a user selected in another widget.

- An integer value that specifies an index in the list of selected devices.

For example, to use an attribute from the last device selected by the user in another widget, you would use the following variable, substituting the attribute name where indicated:

```
{context.deviceObjects.last.<attribute>}
```

The attributes that are available for devices, interfaces, IT services, and organizations are the same as the non-list attributes available for the equivalent ScienceLogic API resource. For example, the attributes available for devices are the same as the attributes for a /device resource in the API. Non-list attributes are attributes that have single assigned value; for example, the child_devices attribute for devices cannot be used as it is a list that can include multiple values.

The following are the most commonly used device attributes:

- **id.** The numeric ID of the device.

- **hostname.** The hostname of the device discovered via hostname discovery.

- **ip.** The IP address the ScienceLogic platform uses to communicate with the device.

- **name.** The name of the device.

- **organization.** The organization of the device. If you use this attribute, you must specify the organization attribute that you want to use. If you use this attribute, you must use the following variable syntax:

```
{context.deviceObjects.<entity index>.organization.<organization attribute>}
```

- Any Base or Extended Custom Attributes that have been added to your ScienceLogic system.

The following are the most commonly used interface attributes:

- **device.** The device with which the interface is associated. If you use this attribute, you must specify the device attribute that you want to use. If you use this attribute, you must use the following variable syntax:

```
{context.interfaceObjects.<entity index>.device.<device attribute>}
```

- **organization.** The organization with which the interface is associated. If you use this attribute, you must specify the organization attribute that you want to use. If you use this attribute, you must use the following variable syntax:

```
{context.interfaceObjects.<entity index>.organization.<organization attribute>}
```

- **ifIndex.** The SNMP index associated with the interface.
• **ifDescr.** The description of the interface.
• **alias.** The alias of the interface.
• **name.** The name of the interface.

The following are the most commonly used IT service attributes:

• **service_id.** The numeric ID of the IT service.
• **service_name.** The name of the IT service.

The following are the most commonly used organization attributes:

• **company.** The name of the organization.
• **billing_id.** The billing ID of the organization.
• **crm_id.** The CRM ID of the organization.

For example, suppose you want to display the **Device Properties** page for the last selected device. In our test system, the URL for the **Device Properties** page for the device with a device ID of “201” is:

   http://10.0.2.59/em7/index.em7?exec=device_details&did=201

We could then edit this URL to specify that instead of the device with the device ID 201, we want to display the **Device Properties** page for the last selected device. We would specify the following:

   http://10.0.2.59/em7/index.em7?exec=device_details&did=
   {context.deviceObjects.last.id}
Viewing the Context Iframe Content Widget

If you specified that you want to view the Device Properties page for the last selected device, the Context Iframe Content widget might look like this:
Embedding a Page from the ScienceLogic Platform in Another Web Page

The web interface provided by Administration Portals, Database Servers, and All-In-One Appliances supports HTTP authentication. HTTP authentication allows you to embed pages from the user interface in an external web portal.

**NOTE:** The examples in this section use secure HTTPS. HTTP authentication can also be used with unsecure HTTP.

To access a ScienceLogic page using HTTP authentication, perform the following:

1. Login to the Administration Portals, Database Servers, and All-In-One Appliances. Navigate to the page you want to embed in the external web page.

2. Copy the URL of the page. For example, suppose you wanted to access a dashboard using HTTP authentication. The URL for a dashboard in the ScienceLogic platform looks like this:


   Where the "X" at the end of the URL is the ID for the dashboard.

3. Append "&use_http_auth=1" to the end of the URL. For example, to request dashboard ID 1 using HTTP authentication, you would use the following URL and supply a username and password in the request:

   `https://ip-address-of-appliance/em7/index.em7?exec=dashboards&dash_id=1&use_http_auth=1`
4. Several pages in the platform allow you to hide the header bar and left NavBar. If you hide the header bar and/or the left NavBar on a page, the URL will change to reflect your choice and can be used to request the page without the header bar and/or the left NavBar.

   For example, if you hide the header bar for a dashboard by selecting the up arrow button:

   ![Dashboard with header bar hidden](image)

   The URL changes to:


   Where 'X' is the ID for the dashboard.

5. To request dashboard ID 1 without the header bar using HTTP authentication, you would use the following URL and supply a username and password in the request:


6. To use the URL of the ScienceLogic page in an HTTP request from an external web page:

   - The URL query string for the ScienceLogic page must include the option "use_http_auth=1".
   - You must pass a ScienceLogic username and password using the standard HTTP authentication mechanism.

---

### Embedding a Single Widget from the ScienceLogic Platform in Another Web Page

The ScienceLogic platform allows you to embed a single widget from a dashboard in an external web page and use HTTP authentication to dynamically update the data in the widget.

To include a single widget in an external web page:

1. Go to the **Dashboards** page. Create a dashboard and configure the widget you want to export, and select any necessary contextual items, so that the widget includes the data you want to display. For details on creating a dashboard and configuring widgets, see the manual [Dashboards](#).
2. Find the widget you want to export. Select the **Options** menu for the widget. Select **Export**.

3. In the **Export Widget** modal page:
   - **Authentication**. Select **HTTP Authentication**.
   - **Export**. Select this button.
   - **Exported URL**. Copy this URL. This URL is unique for this widget, populated with the specified data and with the specified context selected.

4. To use the URL of the widget in an HTTP request from an external web page, you must pass a ScienceLogic username and password using the standard HTTP authentication mechanism.
What is a Proxied Web Service?

Overview

A proxied web service defines how the ScienceLogic platform should perform requests to an external web service. When you define a proxied web service in the ScienceLogic platform, the platform automatically authenticates with the external web service and retrieves data from the external web service so that each user can view the data in a graphical report. The definition of a proxied web service includes:

- The URL of the web service.
- How to pass authentication information to the external web service, so the user does not have to provide a username and password to view information from the web service.
- Optional XSL transformations to apply to the data retrieved from the external web service.
- Other options that must be included in the request, for example, special headers or options.

You can use a proxied web service to create a Proxied Web Service Widget. A proxied web service widget displays the response from the web service in a graphical report in a dashboard. Every time a user views a dashboard that contains a Proxied Web Service Widget, the ScienceLogic platform uses the username, Email address and/or password of that user to authenticate the request to the external web service.
Defining a Proxied Web Service

To use a proxied web service in a dashboard, you must:

1. Create a SOAP/XML credential for the web service.
2. Define the proxied web service in the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services).
3. Optionally, you might want to apply XSL transformations to the data returned by the web service.
4. Optionally, you might want to define a DNS-Based Proxy, to ensure that the ScienceLogic platform can accurately resolve all links in the returned data.
5. To display the data returned by the external web service, you must define a widget in a dashboard page.

Authentication Types

The ScienceLogic platform can use the following methods to authenticate requests to the external web service:

- **HTTP Authentication.** Use this method for web services that authenticate using standard HTTP authentication, such as an Integration Server. The platform can pass a username, Email address, and/or password to the web service.

- **Custom Header.** Use this method for web services that authenticate requests by examining the source IP and a custom HTTP header. For example, the web service for a Splunk server uses this type of authentication. Each time a user views a widget (report in the Dashboards page) that contains dynamic data from Splunk, the ScienceLogic platform builds a custom request header that includes the username, and/or Email address of the user viewing the widget and sends the custom request header to Splunk. To use a custom header, typically you must configure the external web service to include the Administration Portal or All-In-One Appliance in the list of trusted IP addresses.

- **PHP script.** Use this method for web services that cannot be authenticated using the HTTP Authentication or Custom Header methods. The ScienceLogic platform executes a custom PHP script to authenticate requests.

XSL Transformations

In addition to specifying how the ScienceLogic platform should perform requests on a web service, you can also specify an optional XSL transformation to apply to the response. The platform will perform the XSL transformation on the response to generate the output that will be displayed to the user. For the platform to apply an XSL transformation to a response from a proxied web service:

1. You must upload the XSL stylesheet to the ScienceLogic platform in the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).
2. The XSL stylesheet can include references to additional files, such as CSS or XSL template files. However, you must upload each additional file to the platform in the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).

3. You must create an XSL transformation definition in the Proxy XSL Transformations page. An XSL transformation definition specifies:
   - The name the platform will use to refer to the XSL transformation in the user interface.
   - The XSL Stylesheet that will be used to perform the transformation.
   - Any additional files that are used by the XSL Stylesheet.

4. In the Proxied Web Service definition, you can specify one or more XSL transformation definitions that can be applied to the responses from that web service.

5. In the widget definition, the user must select the XSL transformation definition that the platform will use to transform the responses for that instance of the widget. Users can select from the list of allowable XSL transformations specified in the definition for that proxied web service.

**DNS-Based Proxy**

For some web services, the default method the ScienceLogic platform uses to send proxy requests causes URI references in the response to be unresolvable. For example, if the response links to a CSS or JavaScript file outside of the scope of the proxy, those files will not be available when the page is displayed in a dashboard.

If the response from a web service includes URLs that the ScienceLogic platform cannot resolve using the default proxy method, you can configure a DNS-Based Proxy. Configuring a DNS-Based Proxy allows the platform to resolve URLs that cannot be resolved using the default method. However, to use a DNS-Based Proxy, you must define a *wildcard DNS CNAME record* that:

- Resolves to the Administration Portal or All-In-One Appliance that the user is connecting to.
- Can be resolved by the system running the client browser.

When you define a wildcard CNAME record for the Administration Portal or All-In-One Appliance, the URL for those servers will resolve to the Administration Portal regardless of the text at the front of the URL. During requests to proxied web services, the ScienceLogic platform then creates a custom URL for the Administration Portal. The platform inserts information about the current session (user ID and browser session) into the front of the URL for the Administration Portal. The custom URL allows the platform to resolve absolute URLs even when the request doesn't include a referer header.

To configure DNS-Based Proxy, you must define rules in the DNS-Based Proxy Rules page (Registry > Web Proxies > DNS-Based Proxy). If your ScienceLogic system includes multiple Administration Portals or if different CNAME records apply to different users based on their network location, you can define multiple DNS-Based Proxy rules.

A DNS-Based Proxy rule defines:

- The users who will use this DNS-Based Proxy rule, based on the address the user supplied to connect to the Administration Portal or All-In-One Appliance.
- The order in which the platform should use the DNS-Based proxy rules.

What is a Proxied Web Service?
- The domain specified in the DNS CNAME record.
- The protocol (HTTP or HTTPS) to use. You can also specify that you want to disable the protocol. If the protocol is disabled, the platform will use the default, non DNS-Based, proxy method for users that match this DNS-Based proxy rule.
Chapter 7

Creating a Credential for a Proxied Web Service

Overview

Before creating a proxied web service, you must create a credential for that web service. Proxied Web Services use SOAP/XML credentials. Before creating a credential, however, you must determine how the ScienceLogic platform will authenticate requests to the external web service. The platform can use the following authentication methods:

- **HTTP Authentication.** Use this method for web services that authenticate using standard HTTP authentication, such as an Integration Server. The ScienceLogic platform can pass a username, Email address and/or password to the web service.

- **Custom Header.** Use this method for web services that authenticate the user if the request comes from a trusted IP address and a custom HTTP header for the request includes specific user information. For example, you can use this type of authentication to retrieve data from a Splunk server. The ScienceLogic platform authenticates by including the username, and/or Email address of the user viewing the proxied web service widget in the request headers. Typically, you must configure the external web service to include the Administration Portal or All-In-One Appliance in the list of trusted IP addresses.

- **PHP script.** Use this method for web services that cannot be authenticated using the **HTTP Authentication** or **Custom Header** methods. The ScienceLogic platform executes a custom PHP script to authenticate requests.

This chapter will discuss how to create a credential for each type of authentication method.
Creating a Credential for HTTP Authentication

You can use HTTP Authentication for web services that authenticate using standard HTTP authentication, such as an Integration Server. The ScienceLogic platform can pass the username, Email address and/or password of the user viewing the proxied web service widget to the web service.

To create a credential for a proxied web service using the HTTP authentication method:

1. Go to the Credential Management page (System > Manage > Credentials).
2. In the Credential Management page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
3. The Credential Editor page appears. The ScienceLogic platform will use the following fields when using the SOAP/XML credential for a proxied web service using HTTP Authentication:

   - **Profile Name.** Name of the profile. Can be any combination of alphanumeric characters.
   - **URL.** If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the Credential Editor page requires a value in this field.
   - **HTTP Auth User.** User name with which to log in to the web server. You can include one or more of the following variables in this field:
     - %u. Username of the user currently logged in to the Administration Portal or All-In-One Appliance.
     - %e. Email address of the user currently logged in to the Administration Portal or All-In-One Appliance.
• **HTTP Auth Password.** Password with which to access the web server. You can include the following variable in this field:
  - `%p`. Password of the user currently logged in to the Administration Portal or All-In-One Appliance.

• **Proxy Settings.** This is an optional field. If you use a proxy server in front of the web server(s) you want to communicate with, enter values in these fields:
  - **IP.** IP address of the proxy server.
  - **Port.** Port on the proxy server to which you will connect.
  - **User.** Username to use to access the proxy server.
  - **Password.** Password to use to access the proxy server.

• **Curl Options.** This is an optional field. You can include cURL options in your credential. This field includes the list of cURL options you can include in your credential. To include a cURL option in the credential, select it and then select the right-arrow icon. You can then supply arguments in the field to the right of the option.

• **HTTP Headers.** If you require custom HTTP headers to communicate with the web server, you can build the custom header here. However, if you are using the HTTP Authentication method to connect to an external web service, you cannot use substitution characters in this field.

### Creating a Credential for Custom Header Authentication

You can use Custom Header authentication for web services that authenticate the user if the request comes from a trusted IP address and if the request includes a custom HTTP header that contains specific user information.

To create a credential for a proxied web service using the Custom Header authentication method:

1. Go to the **Credential Management** page (System > Manage > Credentials).
2. In the **Credential Management** page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
3. The Credential Editor page appears. The ScienceLogic platform will use the following fields when using the SOAP/XML credential for a proxied web service using Custom Header Authentication:

- **Profile Name.** Name of the profile. Can be any combination of alphanumeric characters.

- **URL.** If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the Credential Editor page requires a value in this field.

- **HTTP Auth User.** Username with which to access the web server. This field is optional. If you are using the Custom Header authentication method, you cannot use substitution characters in this field.

- **HTTP Auth Password.** Password with which to access the web server. This field is optional. If you are using the Custom Header authentication method, you cannot use substitution characters in this field.

- **Proxy Settings.** This is an optional field. If you use a proxy server in front of the web server(s) you want to communicate with, enter values in these fields:
  - **IP.** IP address of the proxy server.
  - **Port.** Port on the proxy server to which you will connect.
  - **User.** User name to use to access the proxy server.
  - **Password.** Password to use to access the proxy server.

- **Curl Options.** This is an optional field. You can include cURL options in your credential. This field displays a list of all the cURL options you can include in your credential. To include a cURL option in the credential, select it and then select the right-arrow icon. You can then supply arguments in the field to the right of the option.
- **HTTP Headers.** If you require custom HTTP headers to communicate with a web server, you can build the custom header here. For a credential that you will use with the Custom Header Authentication option, you can include one or more of the following variables in this field:
  - `%u`. Username of the user currently logged in to the Administration Portal or All-In-One Appliance.
  - `%e`. Email address of the user currently logged in to the Administration Portal or All-In-One Appliance.
  - `%p`. Password of the user currently logged in to the Administration Portal or All-In-One Appliance.

**Creating a Credential for PHP Script Authentication**

You can use a custom PHP Script to authenticate with web services that cannot be authenticated using the **HTTP Authentication** or **Custom Header** methods. The ScienceLogic platform executes a custom PHP script to supply the required authentication information to the web service.

When you use the PHP Script method for authentication, the ScienceLogic platform performs requests to the proxied web service in the following way:

The ScienceLogic platform constructs a PECL (PHP Extension Community Library) HTTP request using the information provided in the associated SOAP/XML credential and the URL provided in the proxied web service definition.

1. The platform executes the PHP script. The PHP script can define HTTP headers, PECL options, and HTTP authentication settings.
2. The platform adds the information defined by the PHP script to the PECL HTTP request.
3. The platform sends the request to the web service.

To create a credential for a proxied web service using the PHP Script authentication method, you must:

- Define the credential that will provide additional information for authentication.
- Define the PHP script that will generate the authentication information.

To create the PHP script:

1. Go to the **Proxied Web Services** page (Registry > Web Proxies > Proxied Web Services).
2. You must include the PHP script within the definition of the proxied web service that uses the PHP script.

   - **If you have not yet defined the proxied web service**, you can supply only a name and the PHP script. Select the [Create] button to create a new Proxied Web Service.
   - **If you have already defined a proxied web service** and want to use a PHP script during authentication, select the wrench icon (🔧) for the Proxied Web Service.
3. The **Proxy Service Editor** page appears. In this page, supply values in the following fields:

![Proxy Service Editor](image)

**NOTE:** This section describes only the fields related to creating a custom PHP script for authentication. The remaining fields are described in detail in the chapter *Defining a Proxied Web Service*.

- **Service Name.** Name of the proxied web service.
- **SOAP/XML Credential.** Select from a list of all SOAP/XML Credentials. Best practice is to create a credential specifically for this proxied web service. Select the credentials you created earlier in this section.
- **Base URL.** Enter the URL for the web service. This field defaults to the URL defined in the credential you selected in the **SOAP/XML Credential** field; however, you can change the default URL.
- **Authentication Type.** Specifies how the platform should perform authentication with the external web service. Select **PHP Script**.
- **Authentication Script.** Enter the text of the PHP script in this field.

**NOTE:** For an example of an authentication script, see the chapter *Example Using PHP Script for Authentication*. 
When the ScienceLogic platform executes the PHP script, the platform passes the following **PHP variables** to the PHP script:

- `$service_id`. The numeric ID for the proxied web service.
- `$auth_user`. The username of the currently logged in user.

The PHP script can modify the following **PHP arrays**:

- `$headers`. An associative array that the ScienceLogic platform supplies as a parameter for the `setHeaders` function. The `setHeaders` function is used to build the PECL HTTP request and allows you to define a custom header for our request. The array keys in the `$headers` array must be set to the name of the header. Each key must point to a value for that header. For more information about the `setHeaders` function, see [http://www.php.net/manual/en/function.httprequest-setheaders.php](http://www.php.net/manual/en/function.httprequest-setheaders.php).
- `$options`. An associative array that the ScienceLogic platform supplies as a parameter for the `setOptions` function. The `setOptions` function is used to build the PECL HTTP request and allows us to specify options to include in the HTTP request (for example, URL, port, redirects, referrer). For more information about the `setOptions` function, see [http://www.php.net/manual/en/function.httprequest-setoptions.php](http://www.php.net/manual/en/function.httprequest-setoptions.php).

The PHP script can modify the following **PHP variables**:

- `$http_user`. You can assign a value to this variable. If this variable and the `$http_password` variable are both not NULL, the ScienceLogic platform will use the value in this variable (in combination with the value in the variable `$http_password`) to perform HTTP authentication. The platform will use this value to build the `httpauth` option in the `setOptions` function.
- `$http_password`. You can assign a value to this variable. If this variable and the `$http_user` variable are both not NULL, the ScienceLogic platform will use the value in this variable (in combination with the value in the variable `$http_user`) to perform HTTP authentication. The platform will use this value to build the `httpauth` option in the `setOptions` function.

4. Select the [Save] button to save the PHP script.

5. Next you must define the credential that supplies additional values during authentication with the web service.

6. Go to the **Credential Management** page (System > Manage > Credentials).

7. In the **Credential Management** page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
8. The Credential Editor page appears. The ScienceLogic platform will use the following fields when creating the SOAP/XML credential for a proxied web service using HTTP Authentication:

- **Profile Name.** Name of the profile. Can be any combination of alphanumeric characters.

- **URL.** If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the Credential Editor page requires a value in this field.

- **HTTP Auth User.** If you supply a value in this field, the ScienceLogic platform will use the specified user name when building the authentication request. If your PHP script defines a value for the $http_user and $http_password variables, the value in this field will not be used.

- **HTTP Auth Password.** If you supply a value in this field, the ScienceLogic platform will use the specified password when building the authentication request. If your PHP script defines a value for the $http_user and $http_password variables, the value in this field will not be used.

- **Proxy Settings.** These are optional fields. The ScienceLogic platform will include these fields when building the authentication request. If you use a proxy server in front of the web server(s) you want to communicate with, enter values in these fields:
  - **IP.** IP address of the proxy server.
  - **Port.** Port on the proxy server to which you will connect.
  - **User.** Username to use to access the proxy server.
  - **Password.** Password to use to access the proxy server.
- **CURL Options**: This is an optional field. You can include cURL options in your credential. This field displays a list of all the cURL options you can include in your credential. To include a cURL option in the credential, select it and then select the right-arrow icon. You can then supply arguments in the field to the right of the option. The ScienceLogic platform will include these cURL options when building the authentication request.

- **HTTP Headers**: If you require custom HTTP headers to communicate with a web server, you can build the custom header here. However, if you are using the PHP Script Authentication method to connect to an external web service, you cannot use substitution characters in this field.
Chapter 8

Defining Proxied Web Services

Overview

After you have created a credential for the proxied web service, you can create, edit, and delete the definition of the proxy web service. This chapter will describe how to create, view, edit, and delete a proxied web service in the Proxied Web Services page.

Creating a Proxied Web Service

You can create a proxied web service in the Proxied Web Services page. To create a proxied web service:

1. Go to the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services).
2. Select the [Create] button in the upper right of the page. The Proxy Service Editor page appears.
3. The **Proxy Service Editor** page includes the following fields:

- **Service Name.** Name of the proxied web service.
- **SOAP/XML Credential.** Select from a list of all SOAP/XML Credentials. Best practice is to create a credential specifically for this proxied web service. Select the appropriate credential from this list.
- **Base URL.** This value is initially populated using the URL defined in the selected credential. However, you can modify the default value to ensure that this field specifies the URL of the external web service that you want to display in a widget. When the ScienceLogic platform forwards a request from the browser, the request will always start with this URL. Although the user can follow links that include this URL, the platform will not let the user make a request outside this URL. The **Base URL** field enforces access control.
- **Authentication Type.** Specifies how the ScienceLogic platform should perform authentication with the external web service. Choices are:
  - **HTTP Authentication.** Use this method for web services that authenticate using standard HTTP authentication, such as the Integration Server. The ScienceLogic platform can pass the username, Email address, and/or password of the current user (viewing the proxied web service widget) to the web service. The platform uses the user name, Email address, and/or password from the aligned credential.
    - If the **HTTP Auth User** field in the credential and/or the **HTTP Auth Password** field in the credential include substitution variables, the ScienceLogic platform will replace each variable with values from the current session.
  - **Custom Header.** Use this method for web services that authenticate based on:
    - request is sent from a trusted IP address. You must configure the external web service to include the Administration Portal or All-In-One Appliance in the list of trusted IP addresses.
request includes a custom HTTP header. Usually this header must use a header name known to the web service and include a value (for example, the username) known to the web service. This custom header is defined in the aligned credential.

- If the HTTP Headers field in the credential includes substitution variables, the ScienceLogic platform will replace each variable with values from the current session.

For example, the web service for a Splunk server uses this type of authentication. The ScienceLogic platform authenticates by including the username of the current user (who is viewing the proxied web service widget) in a custom request header.

- PHP script. Use this method for web services that cannot be authenticated using the HTTP Authentication or Custom Header option. The ScienceLogic platform executes a custom PHP script to authenticate requests. The script is defined in the Authentication Script field in this page. For details on how the ScienceLogic platform uses this script and the variables that must be defined by the PHP script, see the Creating a Credential for a Proxied Web Service chapter.

- Authentication Options. This pane includes custom headers and custom PHP authentication scripts.
  - HTTP Headers. If the aligned credential includes HTTP Headers, each header appears in a list in this pane.
  - Authentication Script. If you selected PHP Script in the Authentication Type field, enter the PHP script in this field.

- Output Options. If you have defined one or more Proxy XSL Transformations, each Proxy XSL Transformation appears in this field. You can select one more Proxy XSL Transformations to align with this proxied web service. When you align an XSL transformation with a proxied web service, you are indicating to the ScienceLogic platform that the XSL transformation is valid and can be used to transform the response from the web service. The selected XSL transformations will be available as options when a widget is created for the proxied web service. To learn more about XSL Transformations, see the XSL Transformations chapter in this manual.

Viewing the List of Proxied Web Services

The Proxied Web Services page displays a list of existing proxied web services. To view a list of proxied web services:

1. Go to the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services). For each policy, the Proxied Web Services page displays:
Defining Proxied Web Services

• **Service Name.** Name of the Proxied Web Service.
• **Credential.** Name of the credential aligned with the Proxied Web Service.
• **Base URL.** URL and optionally port to which the Proxied Web Service will send requests.
• **Authentication.** Authentication method for the Proxied Web Service. Choices are:
  - HTTP Authentication. To authenticate requests to the web service, the ScienceLogic platform passes the username, Email address, and/or password of the current user using HTTP authentication.
  - Custom Header. To authenticate requests to the web service, the ScienceLogic platform includes a custom header in each request.
  - PHP Script. The ScienceLogic platform executes a custom PHP script to authenticate requests to the web service.
• **Edited By.** User who created or last edited the policy.
• **Last Edited.** Date and time the policy was created or last edited.

### Editing and Deleting a Proxied Web Service

You can edit an existing Proxied Web Service. To do this:

1. Go to the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services).
2. Find the proxied web service you want to edit. Select its wrench icon (🔧).
3. In the Proxy Service Editor modal page, you can edit the value in one or more fields described in the section **Creating a Proxied Web Service**.
4. Select the [Save] button to save your changes to the Proxied Web Service.
5. If you have already created a widget that displays the retrieved data from the Proxied Web Service, the widget will update dynamically to display the changes in the Proxied Web Service.
Deleting a Proxied Web Service

To delete one or more proxied web services:

1. Go to the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services).
2. Find the Proxied Web Service you want to delete. Select its checkbox.
3. Select the checkbox for any more Proxied Web Services you want to delete.

5. Each selected Proxied Web Service is removed from the ScienceLogic platform.
Using the Proxied Web Service Widget

Overview

A dashboard is a page that displays one or more graphical reports, called widgets. The ScienceLogic platform includes pre-defined widget definitions that can be customized and displayed in the Dashboards page.

You can use the proxied web service widget to view data retrieved from an external website.

For more information on Dashboards, see the Dashboards manual.

The Proxied Web Service Widget

You can view the proxied web service widget in the Dashboards page ([Dashboards] tab). To view the proxied web service widget:

2. After you have configured your Dashboard, select the Proxied Web Service widget type.
3. You have the following options when configuring the **Proxied Web Service** widget:

   - **Web Service.** Select from a list of proxied web services in the ScienceLogic platform. Data from the URL specified in the proxied web service will be displayed in the widget. The platform will automatically pass authentication data to the external web service.

   - **Service Base URL.** A read-only file, populated automatically by the ScienceLogic platform when you select a value in the **Web Service** field. This value is the URL of the external web service that you want to display in a widget. When the ScienceLogic platform forwards a request from the browser, the request will always start with this URL. Although the user can follow links that include this URL. The platform will not let the user make a request outside this URL.

   - **Target URL.** This field is optional. If you would like to include an additional directory in the **Service Base URL**, you can specify that in this field. For example, suppose the **Service Base URL** displays the top-level overview page of a web service, like "www.webservice.com/intro". Suppose you always want the widget to display the sub-page "Account Details", like "www.webservice.com/intro/account". You could specify the URL of the sub-page in the **Target URL** field.

   - **Default Transformation.** The ScienceLogic platform allows you to apply an XSL transformation to the data displayed in a widget. Select from the list of allowable XSLT transformations for this proxied web service.

   - **Include white background behind content.** Select this checkbox if you want a white background for your widget.

4. The **Options** menu drop-down in the upper right of the Proxied Web Service widget has the following options:

   - **Show URL.** Displays the URL of the proxied web service.

   - **No Transformation.** The widget will be displayed without XSL Transformations applied.

   - **XSL Transformations.** Will display any possible XSL Transformations for the proxied web service widget.
Depending on your selections, your **Proxied Web Service** widget might look like this:

![Proxied Web Service Widget]

<table>
<thead>
<tr>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get/Update/Add/Delete User Accounts</td>
<td>Account</td>
</tr>
<tr>
<td>Add Alerts</td>
<td>Alert</td>
</tr>
<tr>
<td>Get/Update XMF Appliances</td>
<td>XMF</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Asset Records</td>
<td>Asset</td>
</tr>
<tr>
<td>Get/Update/Add/Delete CP Theme Resources</td>
<td>Theme</td>
</tr>
<tr>
<td>Get/Update/Add Credentials</td>
<td>Credentials</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Dashboards</td>
<td>Dashboard</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Devices and Get Collected Data</td>
<td>Devices</td>
</tr>
<tr>
<td>Get Device Classes</td>
<td>Classes</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Groups</td>
<td>Groups</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Templates</td>
<td>Templates</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Discovery Sessions</td>
<td>Discovery Sessions</td>
</tr>
<tr>
<td>Move/Unmove Active Devices Discovery Sessions</td>
<td>Discovery Sessions</td>
</tr>
<tr>
<td>Get Dynamic Application Resources</td>
<td>Dynamic App</td>
</tr>
<tr>
<td>View/Update/Clear Events</td>
<td>Events</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Monitor Policies</td>
<td>Monitor</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Organizations</td>
<td>Organizations</td>
</tr>
<tr>
<td>Get/Update/Add/Delete Theme Resources</td>
<td>Themes</td>
</tr>
</tbody>
</table>
Overview

You can specify an optional XSL transformation to apply to the response from a web service. The ScienceLogic platform will perform the XSL transformation on the response and generate the output that will be displayed to the user.

For proxied web services, you can use an XSL transformation to prepare the retrieved data for displays in a widget. Some retrieved data requires a transformation before it looks pleasing in a widget.

For the ScienceLogic platform to apply an XSL transformation to a response from a proxied web service:

1. You must upload the XSL stylesheet to the ScienceLogic platform in the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).
2. The XSL stylesheet can include references to additional files, such as CSS or XSL template files. However, you must upload each additional file to the ScienceLogic platform in the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).
3. You must create an XSL transformation definition in the Proxy XSL Transformations page. An XSL transformation definition specifies:
   - The name the platform will use to refer to the XSL transformation in the user interface.
   - The XSL Stylesheet that will be used to perform the transformation.
   - Any additional files that are used by the XSL Stylesheet.
4. In the Proxied Web Service definition, you must specify one or more XSL transformation definitions that can be applied to the responses from that web service.
5. In the widget definition, the user must select the XSL transformation definition that the ScienceLogic platform will use to transform the responses for that instance of the widget. Users can select from the list of allowable XSL transformations specified in the definition for that proxied web service.
This chapter will discuss:

- Uploading Service Files
- Viewing the List of Service Files
- Editing a Service File
- Creating a Proxy XSL Transformation
- Viewing the list of Proxy XSL Transformations
- Editing and Deleting a Proxy XSL Transformation
- Associating an XSL Transformation with a Proxied Web Service
- Using an XSL Transformation with a Proxied Web Service

Uploading Service Files

The Proxy Service Files page allows you to view and upload XSLT stylesheets and associated files to the ScienceLogic platform. You can then use the uploaded files to define a Proxy XSL Transformation.

To upload a proxy service file:

1. Go to the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).
2. In the Proxy Service Files page, select the [Create] button.
3. The File Editor page appears. Define values in the following fields:

   - **File Name.** The name of the Proxy Service File. This name will appear as a choice in the Proxy Service Editor page, in the Service Files Used and XSL Stylesheet fields.
   - **Upload File.** Use this field to select the file you want to upload. To select a file to upload, select the [Browse] button.
- **File Editor.** If you are uploading a new Proxy Service File, leave this field blank. If you are editing an existing Proxy Service File, you can edit the file in this field without uploading the file again.

4. Select the [Save] button to upload the Proxy Service File to the ScienceLogic platform. You can now use the Proxy Service file in the definition of a proxy XSL transformation.

### Viewing The List of Service Files

The **Proxy Service Files** page displays a list of all Proxy Service Files that can be used with proxy XSL transformations. To view the list of Proxy Service Files:

1. Go to the **Proxy Service Files** page: (Registry > Web Proxies > Proxy Service Files).
2. For each Proxy Service File, the **Proxy Service Files** page displays the following:

   - **File Name.** Name of this Proxy Service File.
   - **# Transformations.** Number proxy XSL transformations that use this Proxy Service File.
   - **Edited By.** User who created or last edited this Proxy Service File.
   - **Last Edited.** Date and time this Proxy Service File was created or last edited.

### Editing a Service File

To edit an existing Proxy Service file:

1. Go to the **Proxy Service Files** page (Registry > Web Proxies > Proxy Service Files).
2. In the **Proxy Service Files** page, find the Proxy Service File you want to edit. Select its wrench icon (🔧).
3. In the **File Editor** modal page, edit the value in one or more fields described in the section *Uploading Service Files*.
4. Select the [Save] button to save your changes to the Proxy Service File.
Creating a Proxy XSL Transformation

To define an XSL Transformation for use with a proxied web service, you must first perform the following steps:

- Do not create the XSL Transformation stylesheet in the ScienceLogic platform. You must first retrieve output from the web service, using the same URL you will use in the proxied web service (you might have to perform a login step when you access the service outside of the platform). You must then examine the output and develop the XSL Transformation style sheet outside of the platform.

- When you have finished developing the XSL Transformation stylesheets, you must then upload the stylesheets and associated files to the ScienceLogic platform. You do this through the **Proxy Service Files** page. To learn how to upload a service file, see the section on **Uploading Service Files**.

After performing these steps, you can define a Proxy XSL Transformation in the ScienceLogic platform. To define a Proxy XSL Transformation:

1. Go to the **Proxy XSL Transformations** page (Registry > Web Proxies > Proxy XSL Transformations).
2. In the **Proxy XSL Transformations** page, select the **[Create]** button.
3. The **XSL Transformation Editor** page appears. Define values in the following fields:

   - **Transformation Name**: The name of the XSL Transformation. This name will appear as a choice in the **Proxy Service Editor** page, in the **Output Options** field.

   - **Menu Text**: Another name for the XSL Transformation. This name will appear in the **Widget Configuration** page for a widget of type **Proxied Web Service**, in the **Default Transformation** field. This name will also appear in the **Options** menu in the upper right of each widget in the **Dashboards** page.

   - **Service Files Used**: This field displays a list of all files in the **Proxy Service Files** page. Select each file that is called, included, or referenced from your XSLT stylesheet. You must have already uploaded these files to the **Proxy Service Files** page.

   - **XSL Stylesheet**: This field displays a list of all files in the **Proxy Service Files** page that end with .xsl. Select the XSL stylesheet that defines this XSL Transformation.
4. Select the [Save] button to save the XSL Transformation.
5. You can now use the XSL Transformation in the definition of a proxied web service and for use in a widget in the [Dashboards] tab.

**Viewing the List of Proxy XSL Transformations**

The Proxy XSL Transformations page displays a list of all XSL Transformations that can be used with proxied web services. To view the list of proxy XSL Transformations:

1. Go to the Proxy XSL Transformations page (Registry > Web Proxies > Proxy XSL Transformations).
2. For each XSL Transformation, the Proxy XSL Transformations page displays the following:

- **Transformation name.** Name of this XSL Transformation.
- **Menu Text.** Name of this XSL Transformation as it appears in menus in the [Dashboards] tab.
- **ID.** Unique numeric ID automatically assigned to this XSL Transformation by the ScienceLogic platform.
- **Stylesheet Name.** Name of the XSL stylesheet associated with this XSL Transformation.
- **# Services.** Number of proxied web services that are aligned with this XSL Transformation.
- **# Files.** Number of service files used by this XSL Transformation.
- **Powerpack.** Name of the PowerPack that includes this XSL Transformation.
- **Edited By.** User who created or last edited this XSL Transformation.
- **Last Edited.** Date and time this XSL Transformation was created or last edited.
Editing and Deleting a Proxy XSL Transformation

You can edit an existing Proxy XSL Transformation from the Proxy XSL Transformations page. To edit a Proxy XSL Transformation:

1. Go to the Proxy XSL Transformations page (Registry > Web Proxies > Proxy XSL Transformations).
2. In the Proxy XSL Transformations page, find the XSL Transformation you want to edit. Select its wrench icon.
3. In the XSL Transformation Editor modal page, edit the value in one or more fields described in the section Creating a Proxy XSL Transformation.
4. Select the [Save] button to save your changes to the XSL Transformation.
5. If you have already aligned the XSL Transformation with a widget, the widget will update dynamically to display the changes in the XSL Transformation.

Deleting a Proxy XSL Transformation

You can delete one or more Proxy XSL Transformations from the Proxy XSL Transformations page. To delete a Proxy XSL Transformation:

1. Go to the Proxy XSL Transformations page (Registry > Web Proxies > Proxy XSL Transformations).
2. Find the XSL Transformation you want to delete. Select its checkbox.
3. Select the checkbox for each XSL Transformation you want to delete.
4. Go to the Select Action field in the lower right of the page. Select Delete Transformations. Select the [Go] button.
5. Each selected XSL Transformation is removed from the ScienceLogic platform. The platform does not delete the files used by the XSL Transformation. The files, including the stylesheet, still appear in the Proxy Service Files page.

**Associating an XSL Transformation with a Proxied Web Service**

You can specify one or more XSL transformations in the definition of a proxied web service. When you specify an XSL transformation in the definition of a proxied web service, you are telling the ScienceLogic platform that the XSL transformation is valid for that proxied web service.

To specify an XSLT stylesheet in the definition of a proxied web service:

1. Go to the Proxied Web Services page (Registry > Web Proxies > Proxied Web Services).
2. Find the Proxied Web Service you want to associate with an XSL Transformation. Select its wrench icon (🔧).
3. In the Proxy Service Editor page, select the stylesheet(s) you want to apply to the proxied web service.
4. Select the [Save] button to apply the changes. The selected XSL transformations will be available as options when a widget is created for the proxied web service.

**Using an XSL Transformation with the Proxied Web Service Widget**

If you have aligned one or more XSLT stylesheets to a proxied web service, you can toggle among the XSLT stylesheets in the Proxied Web Service widget. To do so:

2. After you have configured your Dashboard, select the Proxied Web Service widget type.
3. In the **Widget Configuration** page, the following option relates to XSL Transformations:

- **Default Transformation.** The ScienceLogic platform allows you to apply an XSL transformation to the data displayed in a widget. If the author of the proxied web service aligned one or more XSL transformations with the proxied web service, those aligned XSL transformations appear in this field. Select from the list of allowable XSLT transformations for this proxied web service.

4. After you have configured your Proxied Web Service widget, you can apply a different XSL transformation to the data displayed in a widget. Select the **[Options]** menu drop-down to view any other XSL Transformations associated with the selected proxied web service. If the author of the proxied web service aligned one or more XSL transformations with the proxied web service, those aligned XSL transformations appear in this list. Select from the list of allowable XSLT transformations for this proxied web service. You can also select to view the raw data, without any XSL transformation.
To learn more about the Proxied Web Service widget, see the *Using the Proxied Web Service Widget* chapter in this manual.
What is a DNS-Based Proxy?

Some web services might include absolute URLs that refer to files outside of the Base URL specified in the definition of a Proxied Web Service. In this situation, when retrieving data from an external web service, the ScienceLogic platform might return a 404 "file not found" error. For example, if the response from a proxied web service includes links to a CSS or JavaScript file outside of the Base URL, those files will not be available when the page is displayed in a dashboard widget.

When this happens, the ScienceLogic platform will ask the browser to search its stored hierarchy (for the proxied web service) and try again to find the file. However, this method does not work for every web service. For example, certain ajax requests don't send a referrer header, so the platform will not be able to determine which proxied web service generated the "404" error.

To avoid the "file not found" error, you can define a DNS-Based Proxy. A DNS-Based Proxy allows you to define a CNAME record for the Administration Portal or All-In-One Appliance that includes an initial wildcard. This means that URL will resolve to the Administration Portal or All-In-One Appliance regardless of the text at the front of the URL. During requests to proxied web services, the ScienceLogic platform then creates a custom URL for the Administration Portal or All-In-One Appliance. The platform inserts information about the current session (user ID and browser session) into the front of the URL for the Administration Portal or All-In-One Appliance. The custom URL allows the platform to resolve absolute URLs even when the request doesn't include a referrer header.

If your ScienceLogic system includes multiple Administration Portals or if different CNAME records apply to different users based on their network location, you can define multiple DNS-Based Proxy rules.

A DNS-Based Proxy rule specifies:

- The users to whom this DNS-based proxy rule applies, based on the address the user supplied to connect to the Administration Portal or All-In-One Appliance.
- The order in which the ScienceLogic platform should use the DNS-Based proxy rules.
• The wildcard domain (specified in the DNS CNAME record) that allows the ScienceLogic platform to resolve absolute URLs in the proxied web service.
• The protocol (HTTP or HTTPS) to use. If you specify “Disabled”, users who match the rule will not use DNS-based proxy.

This chapter includes details on:

• Viewing a List of DNS-Based Proxies
• Prerequisites for Defining a DNS-Based Proxy
• Defining a DNS-Based Proxy
• Editing a DNS-Based Proxy
• Deleting a DNS-Based Proxy

Viewing a List of DNS-Based Proxies

The DNS-Based Proxy Rules page displays a list of all DNS proxy rules that are defined in the ScienceLogic platform. To view the list of DNS-Based Proxy Rules:

1. Go to the DNS-Based Proxy Rules page (Registry > Web Proxies > DNS-Based Proxy).

2. For each DNS-Based Proxy Rule, the DNS-Based Proxy Rules page displays the following:
   
   • AP Hostname Pattern. Specifies the users to whom this DNS-based proxy rule applies. When a user accesses the Administration Portal or All-In-One Appliance using this pattern, the ScienceLogic platform will apply the DNS-based proxy rule when executing a proxied web service.
   
   • Type. Method the ScienceLogic platform uses to match the AP Hostname Pattern against session information on the Administration Portal or All-In-One Appliance. Choices are "Wildcard" or "Regex".
   
   • Order. If there are multiple DNS-based proxy rules, precedence for this rule.
• **ID.** Unique numeric ID automatically assigned to this DNS-based proxy rule by the ScienceLogic platform.

• **Wildcard Domain.** The wildcard domain that will replace the standard IP address or URL of the Administration Portal or All-In-One Appliance during requests to the proxied web service. This domain always begins with "*.em7.proxy."

• **Protocol.** Protocol the ScienceLogic platform will use to contact the wildcard domain. Choices are: HTTP, HTTPS, and Disabled. If you specify "Disabled", users who mach the rules will not use DNS-based proxy.

• **Edited By.** User who created or last edited this DNS-based proxy rule.

• **Last Edited.** Date and time this DNS-based proxy rule was created or last edited.

---

### Prerequisites for Defining a DNS-Based Proxy

Before you can define a DNS-based proxy rule in the ScienceLogic platform, you must perform the following:

1. Create a **CNAME** entry in your DNS system.

2. This domain for this CNAME entry must start with ".*". This new domain, called the wildcard domain, can simply be your normal domain for the Administration Portal, Database Server, or All-In-One Appliance preceded by ".*".

3. For example, if the URL of your Administration Portal is "MyAdministrationPortal.com", you might enter "*.MyAdministrationPortal.com"

4. The CNAME entry must map to the IP address of the Administration Portal, Database Server, or All-In-One Appliance.

5. The browser for each user who views the data from the proxied web service must be able to access the DNS CNAME entry.

---

### Defining a DNS-Based Proxy

After performing the steps in the **Prerequisites**, you can define a DNS-based proxy rule, using the newly defined wildcard domain. To define a DNS-based proxy rule:

1. Go to the **DNS-Based Proxy Rules** page (Registry > Web Praxies > DNS-Based Proxy).

2. In the **DNS-Based Proxy Rules** page, select the [Create] button.

3. The **DNS Proxy Rule Editor** modal page appears:
4. In the DNS Proxy Rule Editor modal page, supply a value in the following fields.

- **Pattern Type.** Specifies how the ScienceLogic platform will evaluate the value in the **AP Hostname Pattern** field. Choices are:
  - **Wildcard.** The platform will perform a text match, with wildcard characters.
  - **Regex.** The platform will use regular expressions to compare the AP Hostname Pattern to the current session information.

- **AP Hostname Pattern.** This field allows the ScienceLogic platform to examine sessions on the Administration Portal or All-In-One Appliance and determine if the platform should apply the DNS-based proxy rule to the session. Specifically, the platform examines the URL or IP address that the user specified in the browser to connect to the Administration Portal or All-In-One Appliance. If the URL or IP address matches the value in this field, the platform applies the DNS-based proxy rule to the current session.
  - For example, if we specified "*" (asterisk), any IP address or URL will match. The platform will then use this DNS-based proxy rule for every session on the Administration Portal or All-In-One Appliance.
  - If we entered "192.168.38.235", each session on the Administration Portal or All-In-One Appliance where the user enters "192.168.38.235" into the browser will match this DNS-based proxy rule.
  - If we entered ".sciencelogic.local", each session on the Administration Portal or All-In-One Appliance where the user enters ".sciencelogic.local" into the browser will match this DNS-based proxy rule.

- **Priority Order.** If your ScienceLogic system includes multiple DNS-based proxy rule, the platform evaluates the rules in priority order, ascending. The first rule that matches the current session on the Administration Portal or All-In-One Appliance will be used during the current session.

- **Wildcard Domain.** Enter the wildcard domain you defined in the DNS CNAME record for the Administration Portal or All-In-One Appliance, excluding the "*." from the URL. For example, if your DNS CNAME record specifies ".MyAdministrationPortal.com", enter "MyAdministrationPortal.com. When a session on the Administration Portal or All-In-One Appliance matches the **AP Hostname Pattern**, the platform will substitute this wildcard domain into the URL of the Administration Portal or All-In-One Appliance during requests to the proxied web service. The wildcard domain always begins with "*.em7.proxy.", which is added automatically by the platform.
- **Browser Protocol.** Protocol the ScienceLogic platform will use to contact the wildcard domain. Choices are:
  - HTTP
  - HTTPS
  - Disabled. If you select this option, sessions that match the AP Hostname Pattern will **not** use DNS-based proxy.

5. Select the [Save] button to save your new DNS-based proxy rule.

6. Each time a user access the Administration Portal or All-In-One Appliance in a manner that matches the new DNS-based proxy rule, the ScienceLogic platform will apply the DNS-based proxy rule. The platform will substitute the wildcard domain for the URL of the Administration Portal or All-In-One Appliance service when making requests to the proxied web service.

---

**Editing a DNS-Based Proxy**

To edit an existing DNS-based proxy rule:

1. Go to the **DNS-Based Proxy Rules** page (Registry > Web Proxies > DNS-Based Proxy).

2. In the **DNS-Based Proxy Rules** page, find the rule you want to edit. Select its wrench icon (🔧).

3. The **DNS Proxy Rule Editor** modal page appears.

4. In the **DNS Proxy Rule Editor** modal page, edit the values in one or more fields. For details on each field, see the section on **defining a DNS-based proxy**.

5. Select the [Save] button to save your changes to the DNS-based proxy rule.

---

**Deleting a DNS-Based Proxy**

To delete one or more DNS-based proxy rules:

1. Go to the **DNS-Based Proxy Rules** page (Registry > Web Proxies > DNS-Based Proxy).
2. In the DNS-Based Proxy Rules page, find the rule you want to delete. Select its checkbox (✓).

3. Select the checkbox for each rule you want to delete.


5. Each selected rule is removed from the ScienceLogic platform.
Overview

This chapter will walk you through an example web proxy that requests data from the web interface of a Splunk server. The ScienceLogic platform then displays the top-level page of the Splunk web interface in a Dashboard widget.

The example in this chapter uses a Custom Header for authentication. The ScienceLogic platform will send a custom header to the web service for Splunk. Splunk will examine the custom header and the originating IP address (the IP address of the Administration Portal or All-In-One Appliance) to authenticate the ScienceLogic platform.

To create this example, we will:

1. Configure Splunk to accept requests from the ScienceLogic platform.
2. Test the request.
3. Create a credential.
4. Define a DNS-based Proxy.
5. Define a Proxy Web Service in the ScienceLogic platform.
6. Define a Dashboard Widget to display the results of the request.
Configuring Splunk to Accept Requests from the ScienceLogic platform

NOTE: To download a trial copy of Splunk, go to http://www.splunk.com/download. Download the version that is appropriate for your operating system. Install and configure according to Splunk’s instructions.

To allow Splunk to authenticate the ScienceLogic platform using the custom header option, edit the following files on the Splunk Server:

1. Go to /opt/splunk/etc/system/local/web.conf and open with a text editor (like vi).
2. We have to tell Splunk the name of the custom header we will be sending (Remote_User). Find the following section and edit it to look like this:

   # Remote user HTTP header sent by the authenticating proxy server.
   # This header should be set to the authenticated user.
   #
   remoteUser = Remote_User

3. We have to tell Splunk the IP address of the Administration Portal or All-In-One Appliance. Find the following section and edit it to look like this:

   # Trusted IP. This is the IP address of the authenticating proxy.
   # Splunkweb verifies it is receiving data from the proxy host for all
   # SSO requests.
   # Uncomment and set to a valid IP address to enable SSO.
   #
   trustedIP = 127.0.0.1
   trustedIP = enter the IP address of your All-In-One appliance or your
   Administration Portal We entered 10.100.100.10

4. In the [settings] section of the file, add or change the following settings to look like this:

   x_frame_options_sameorigin = False
   SSOMode = permissive

5. Save your changes and exit the file (in vi, :wq).
6. Go to /opt/splunk/etc/system/local/server.conf and open with a text editor (like vi).
7. Tell Splunk to trust itself. Find the following section and edit it to look like this:

   [general]
   guid = This value will be unique to your Splunk server. Don’t edit this entry.
   serverName = splunk
   trustedIP = 127.0.0.1
8. Save your changes and exit the file (in vi, :wq).

9. For more information on configuring Splunk to use single sign-on and authenticate using a custom header, see:


Testing the Request

To test the request to the web interface of Splunk:

1. Open a browser session. Enter the following in the URL bar:

   http://URL or IP address of the Splunk server:8000

   We entered:

   http://10.100.100.22:8000

2. When prompted, enter your username and password. Your username and password for the Splunk web interface must be the same as your username and password.

3. You should see something like this:
Creating a Credential

Next, we must create a credential for our example. Our example uses a custom header for authentication. To create a credential that provides the information for this type of authentication:

1. Go to the Credential Management page (System > Manage > Credentials).
2. In the Credential Management page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
3. The Credential Editor page appears.

![Credential Editor](image)

4. In the Credential Editor page, supply values in the following fields. If a field is not specified, the ScienceLogic platform does not use that field when authenticating a proxied web service. You can accept the default values for fields that are not specifically described in this section. The platform will use the following fields when creating the SOAP/XML credential for a proxied web service using custom-header authentication:

   - **Profile Name**. Name of the profile. We entered Splunk.
   - **URL**. If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the Credential Editor page requires a value in this field. We entered the URL for the web interface to our Splunk server, https://10.100.100.22:8000.
   - **HTTP Auth User**. We did not enter a value in this field. The ScienceLogic platform will authenticate using the custom header and the IP address of the Administration Portal or All-In-One Appliance.
• **HTTP Auth Password.** We did not enter a value in this field. The ScienceLogic platform will authenticate using the custom header and the IP address of the Administration Portal or All-In-One Appliance.

• **Proxy Settings.** This is an optional field. We didn’t enter any values in these fields.

• **Curl Options.** This is an optional field. We didn’t enter any values in this field.

• **HTTP Headers.** We clicked on the link for Add a Header and defined the following customer HTTP header:

```
Remote_User: %u
```

  • Remote_User is the name of the custom HTTP header configured in Splunk.
  • %u is a variable. The ScienceLogic platform will substitute the user name of the user currently logged in to the Administration Portal or All-In-One Appliance.

**TIP:** If your Splunk system does not include the same user account information as the ScienceLogic platform, you could configure the ScienceLogic platform to use a single user account for authentication by replacing %u with the username of that account.

5. Select the **[Save]** button to save the new credential.

## Defining a DNS-based Proxy

To communicate with the web interface for a Splunk server, you must first create a DNS-based proxy. This is because the Splunk server includes links that can’t be resolved by the Administration Portal or All-In-One Appliance.

A DNS-based proxy ensures that the Administration Portal or All-In-One Appliance will be able to resolve links embedded in the results of a request to an external web service.

A DNS-Based Proxy requires a CNAME record for the Administration Portal or All-In-One Appliance. This CNAME record includes an initial wildcard. This means that URL will resolve to the Administration Portal or All-In-One Appliance regardless of the text at the front of the URL.

During requests to proxied web services, the ScienceLogic platform then creates a custom URL for the Administration Portal or All-In-One Appliance. The platform inserts information about the current session (user ID and browser session) into the front of the URL for the Administration Portal.

First, you must create a CNAME record for your Administration Portal or the All-In-One Appliance that will send requests to the Splunk server.
NOTE: If your ScienceLogic system includes multiple Administration Portals or if different CNAME records apply to different users based on their network location, you can define multiple DNS-Based Proxy rules. For guidance, consult your DNS administrator.

The format of the CNAME record should be:

```
*.domain name of the Administration Portal or All-In-One Appliance IN CNAME domain name of the Administration Portal or All-In-One Appliance.
```

For example, if your Administration Portal has a domain name of "MyAdministrationPortal.com", we could create the following CNAME record:

```
```

Next, you must define the DNS-based proxy in the ScienceLogic platform. To do this:

1. Go to the DNS-Based Proxy Rules page (Registry > Web Proxies > DNS-Based Proxy).
2. In the DNS-Based Proxy Rules page, select the [Create] button.
3. The DNS Proxy Rule Editor modal page appears:

4. In the DNS Proxy Rule Editor modal page, supply a value in the following fields.
   - **Pattern Type.** We selected Wildcard.
   - **AP Hostname Pattern.** We entered * (asterisk), meaning "use the DNS-based proxy rule for every session on the Administration Portal or All-In-One Appliance". The ScienceLogic platform examines the URL or IP address that the user specified in the browser to connect to the Administration Portal or All-In-One Appliance. If the URL or IP address matches the value in this field, the platform applies the DNS-based proxy rule to the current session.
   - **Priority Order.** We specified 1. This DNS-based proxy rule will be applied before other matching rules.
**Wildcard Domain.** Enter the wildcard domain you defined in the DNS CNAME record for the Administration Portal or All-In-One Appliance, excluding the "*.". We entered the domain for our Administration Portal (doc-ap.qa.sciencelogic.local). When a session on the Administration Portal or All-In-One Appliance matches the **AP Hostname Pattern**, the platform will substitute this wildcard domain into the URL of the Administration Portal or All-In-One Appliance during requests to the proxied web service. The wildcard domain always begins with "*.em7proxy.", which is automatically included by the platform.

**Browser Protocol.** Select HTTP.

5. Select the [Save] button to save your new DNS-based proxy rule.

6. Each time a user access the Administration Portal or All-In-One Appliance in a manner that matches the new DNS-based proxy rule, the platform will apply the DNS-based proxy rule. The platform will substitute the wildcard domain for the URL of the Administration Portal or All-In-One Appliance service when making requests to the proxied web service.

### Defining a Proxy Web Service in the ScienceLogic platform

You can create a proxied web service in the **Proxied Web Services** page. To create a proxied web service:

1. Go to the **Proxied Web Services** page (Registry > Web Proxies > Proxied Web Services).
2. Select the [Create] button in the upper right of the page.
3. The **Proxy Service Editor** page appears.
4. In the **Proxy Service Editor** page, supply a value in the following fields:

   - **Service Name.** Name of the proxied web service. We entered Splunk.
   - **SOAP/XML Credential.** We selected the credential we created in the *previous section*. We selected Splunk.
   - **Base URL.** We entered the URL for the web interface of our Splunk server. If you are using your own Splunk server with this example, enter the URL of the web interface for your Splunk server.
   - **Authentication Type.** We selected Custom header. The ScienceLogic platform will pass the user name in a custom header. Splunk will examine this custom header and the IP address of the Administration Portal (or All-In-One Appliance) to authenticate the request.
   - **Authentication Options.** Because we selected Splunk min the SOAP/XML Credential field, the ScienceLogic platform included our custom HTTP header from that credential.
   - **Output Options.** We did not select a Proxy XSL Transformation.

5. Select the **[Save]** button to save the new web proxy service.

**Define a Dashboard Widget to Display the Results**

You can use the *proxied web service widget* to view data retrieved from an external website. You can view the proxied web service widget in the *Dashboards* page ([Dashboards] tab).

To view data from our example proxied web service:

1. Go to the *Dashboards* page ([Dashboards] tab).
2. After you have configured your Dashboard and added a widget, select the **Proxied Web Service** widget type in the left NavBar of the *New Widget Configuration* page.
3. In the **Widget Configuration** page, supply values in the following fields:

   - **Widget Type**. We selected Proxied Web Service.
   - **Widget Title**. We entered Integration Service.
   - **Widget Refresh**. Accept the default value.
   - **Web Service**. Select the proxied web service we defined in the previous section. We selected Splunk.
   - **Service Base URL**. We accepted the default value.
   - **Target URL**. We entered the base URL for the web interface to our Splunk server. We entered https://10.100.100.22:8000. If you would like to include an additional directory in the **Service Base URL**, you can specify that in this field. For example, suppose the **Service Base URL** displays the top-level overview page of a web service, like "www.webservice.com/intro". Suppose you always want the widget to display the sub-page "Account Details", like "www.webservice.com/intro/account". You could specify the URL of the sub-page in the **Target URL** field.
   - **Default Transformation**. We did not select an XSL transformation.
   - **Include white background behind content**. We did not select this checkbox.

4. The widget will display the following:

![Splunk Dashboard Screenshot](image)

   You can click on the links in the widget, and the widget will display the next page of data.
Chapter 13

Example Using HTTP Authentication

Overview

This chapter will walk you through an example web proxy that requests data from the Integration Server and displays the requested data in a Dashboard widget.

The example in this chapter uses **HTTP Authentication**. The ScienceLogic platform will use standard HTTP authentication and pass a username and password to the web service on the Integration Server.

To create this example, we will:

1. **Test the request**.
2. **Create an XSL Transformation**.
3. **Create a credential**.
4. **Define a Proxy Web Service in the ScienceLogic Platform**.
5. **Define an XSL Transformation in the ScienceLogic Platform**.
6. **Define a Dashboard Widget to display the results of the request**.

Testing the Request

In our example, we are going to send a request to the Integration Server.

Our Integration Server resides at IP address 10.100.100.16. To run this example on your Integration Server, substitute the IP address or URL of your Integration Server.

1. We opened a browser session and entered the following in the URL bar:

   https://10.100.100.16
2. We were prompted to enter the user name and password for the Integration Server. We did so.

3. Here is the result of our request to the top-level of the Integration Server:

```
<APIFeatures elmentype="list">
  <link URI="/account" description="Get/Update/Add/Delete User Accounts" elmentype="href"/>
  <link URI="/alert" description="Add Alerts" elmentype="href"/>
  <link URI="/appliance" description="Get/Update EAM Appliances" elmentype="href"/>
  <link URI="/asset" description="Get/Update/Add/Delete Asset Records" elmentype="href"/>
  <link URI="/top_theme" description="Get/Update/Add/Delete CE Theme Resources" elmentype="href"/>
  <link URI="/user-profile" description="Get/Update/Add Credentials" elmentype="href"/>
  <link URI="/dashboard" description="Get/Update/Delete Dashboards" elmentype="href"/>
  <link URI="/device" description="Get/Update/Add/Delete Devices and Get Collected Data" elmentype="href"/>
  <link URI="/device_class" description="Get Device Classes" elmentype="href"/>
  <link URI="/device_group" description="Get/Add/Delete Device Groups" elmentype="href"/>
  <link URI="/device_template" description="Get/Update/Add/Delete Device Templates" elmentype="href"/>
  <link URI="/discovery_session" description="View/Start/Stop Active Device Discovery Sessions" elmentype="href"/>
  <link URI="/discovery_session_active" description="View/Start/Stop Active Device Discovery Sessions" elmentype="href"/>
  <link URI="/dynamic_app" description="Get Dynamic Application Resources" elmentype="href"/>
  <link URI="/event" description="View/Update/Clear Events" elmentype="href"/>
  <link URI="/monitor" description="Get/Update/Add/Delete Monitor Policies" elmentype="href"/>
  <link URI="/organization" description="Get/Add/Delete Organizations" elmentype="href"/>
  <link URI="/theme" description="Get/Add/Delete Theme Resources" elmentype="href"/>
  <link URI="/ticket" description="Get/Update/Add/Delete Ticket" elmentype="href"/>
  <link URI="/ticket_queue" description="Get Ticket Queues" elmentype="href"/>
  <link URI="/ticket_state" description="Get/Update/Add/Delete Custom Ticket States" elmentype="href"/>
  <link URI="/vendor" description="Get/Update/Add/Delete Vendor Records" elmentype="href"/>
</APIFeatures>
```

4. To specify one of the links within this result, we opened a browser session and entered the following in the URL bar:

https://10.100.100.16/account

5. Here is the result:

```
<accounts>
  <searchspec>
    <XML fields that are not relevant to our example>
  </searchspec>
  <total_matched>15</total_matched>
  <total_returned>15</total_returned>
  <result_set elmentype="list">
    <link URI="/account/28" description="acmeuser" elmentype="href"/>
    <link URI="/account/32" description="bcorgan" elmentype="href"/>
    <link URI="/account/19" description="dashboard" elmentype="href"/>
    <link URI="/account/22" description="dashboard_admin" elmentype="href"/>
    <link URI="/account/26" description="dkahane" elmentype="href"/>
    <link URI="/account/1" description="em7admin" elmentype="href"/>
    <link URI="/account/33" description="jproctor" elmentype="href"/>
    <link URI="/account/35" description="kgibson" elmentype="href"/>
    <link URI="/account/34" description="mwashington" elmentype="href"/>
    <link URI="/account/30" description="prevere" elmentype="href"/>
    <link URI="/account/3" description="qauser" elmentype="href"/>
    <link URI="/account/31" description="samadams" elmentype="href"/>
    <link URI="/account/27" description="sjohnson" elmentype="href"/>
  </result_set>
</accounts>
```

---

Example Using HTTP Authentication
Creating an XSL Transformation

Without an XSL transformation, our results are not very usable in a Dashboard Widget.

- The information is not laid out in an easily viewable table.
- There are not clickable hyperlinks but instead hrefs to other pages of data.

To convert the results of our request into a table that includes clickable hyperlinks, we wrote the following XSLT stylesheet. This stylesheet converts the top-level results (APIFeatures) into a table format, with clickable hyperlinks. If a user selects a link in the top-level results, this stylesheet converts the new page of results (result_set) into a table format, with clickable hyperlinks.

We saved this XSLT stylesheet in a file called IS_proxy_xslt.xsl.

**NOTE:** For details on XSLT stylesheets, see the website [http://www.w3schools.com/xsl/xsl_intro.asp](http://www.w3schools.com/xsl/xsl_intro.asp). If you are already familiar with XSLT, you can use this XSLT stylesheet to transform the output from your Integration Server.

```xml
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:template match="/">
    <html>
      <body>
        <xsl:choose>
          <xsl:when test="APIFeatures">
            <table bgcolor="#FFFFFF">
              <tr>
                <th>Description</th>
                <th>Link</th>
              </tr>
              <xsl:for-each select="APIFeatures/link">
                <tr>
                  <td>
                    <xsl:value-of select="@description" />
                  </td>
                  <td>
                    <a>
                      <xsl:attribute name="href"><xsl:value-of select="substring(@URI,2)" /></xsl:attribute>
                    </a>
                  </td>
                </tr>
              </xsl:for-each>
            </table>
          </xsl:when>
          <xsl:when test="/result_set">
            (when a link is selected)
            <table bgcolor="#FFFFFF">
              <tr>
                <!-- Table content here -->
              </tr>
            </table>
          </xsl:when>
        </xsl:choose>
      </body>
    </html>
  </xsl:template>
</xsl:stylesheet>
```
Creating a Credential

Next, we must create a credential for our example. Our example uses standard HTTP authentication. To create a credential that provides the information for standard HTTP authentication:

1. Go to the Credential Management page (System > Manage > Credentials).
2. In the Credential Management page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
3. The Credential Editor page appears.
4. In the Credential Editor page, supply values in the following fields. If a field is not specified, the ScienceLogic platform does not use that field when authenticating a proxied web service. You can accept the default values for fields that are not specifically described in this section. The platform will use the following fields when creating the SOAP/XML credential for a proxied web service using HTTP Authentication:

- **Profile Name**: Name of the profile. We entered Proxy IS.
- **URL**: If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the Credential Editor page requires a value in this field. We entered the URL for our Integration Server, https://10.100.100.16.
- **HTTP Auth User**: Username with which to log in to the web server. We entered %u.
  - %u: Username of the user currently logged in to the Administration Portal or All-In-One Appliance.
- **HTTP Auth Password**: Password with which to access the web server. We entered %p in this field.
  - %p: Password of the user currently logged in to the Administration Portal or All-In-One Appliance.
- **Proxy Settings**: This is an optional field. We didn't enter any values in these fields.
- **Curl Options**: This is an optional field. We didn't enter any values in this field.
- **HTTP Headers**: We did not include a custom HTTP header in this credential.

5. Select the [Save] button to save the new credential.
Defining an XSL Transformation in the ScienceLogic Platform

To define an XSL Transformation in the ScienceLogic platform, we must perform two steps:

1. Upload the stylesheet and any dependent files to the ScienceLogic platform.
2. Create a definition of the XSL Transformation in the ScienceLogic platform, referencing the uploaded files.

This section will explain how to perform these steps for our example Web Proxy.

Remember that in an earlier section, we created an XSLT stylesheet named IS_proxy_xslt.xsl. We must now upload the stylesheet to the ScienceLogic platform:

1. Make sure that the file IS_proxy_xslt.xsl is available from your local computer.
2. Go to the Proxy Service Files page (Registry > Web Proxies > Proxy Service Files).
3. In the Proxy Service Files page, select the [Create] button.
4. The File Editor page appears.
5. In the **File Editor** page, define values in the following fields:

   - **File Name**. We specified the name of our XSLT stylesheet, `IS_proxy_xslt.xsl`. This name will appear as a choice in the **Proxy Service Editor** page, in the **Service Files Used** and **XSL Stylesheet** fields.
   - **Upload File**. Use this field to navigate to the stylesheet.
   - **File Editor**. If you are uploading a new Proxy Service File, leave this field blank. If you are editing an existing Proxy Service File, you can edit the file in this field without uploading the file again.

6. Select the **[Save]** button to upload the Proxy Service File to the ScienceLogic platform.

7. You can now use the Proxy Service file in the definition of a proxy XSL transformation.

8. To define the XSL transformation in the ScienceLogic platform, go to the **Proxy XSL Transformations** page (Registry > Web Proxies > Proxy XSL Transformations).

9. In the **Proxy XSL Transformations** page, select the **[Create]** button.

10. The **XSL Transformation Editor** page appears.

11. In the **XSL Transformation Editor** page, enter values in the following fields:

   - **Transformation Name**. We entered `Simple IS XSL`. This name will appear as a choice in the **Proxy Service Editor** page, in the **Output Options** field.
   - **Menu Text**. Another name for the XSL Transformation. We entered `Simple IS`. This name will appear in the **Widget Configuration** page for a widget of type **Proxied Web Service**, in the **Default Transformation** field. This name will also appear in the **Options** menu in the upper right of each widget in the **Dashboards** page.
   - **Service Files Used**. Because our XSL transformation did not call, include, or reference any additional files, we did not select any files in this field.
   - **XSL Stylesheet**. We selected the stylesheet we uploaded earlier in this section. We selected `IS_proxy_xslt.xsl`.
12. Select the [Save] button to save the XSL Transformation. This XSL Transformation can now be applied from within the ScienceLogic platform.

**Defining a Proxy Web Service in the ScienceLogic Platform**

You can create a proxied web service in the **Proxied Web Services** page. To create a proxied web service:

1. Go to the **Proxied Web Services** page (Registry > Web Proxies > Proxied Web Services).
2. Select the [Create] button in the upper right of the page. The **Proxy Service Editor** page appears.
3. The **Proxy Service Editor** page includes the following fields:

   ![Proxy Service Editor](image)

   - **Service Name**. Name of the proxied web service. We entered Doc IS.
   - **SOAP/XML Credential**. We selected the credential we created in the previous section. We selected Proxy IS.
   - **Base URL**. We entered the URL for our Integration Server. If you are using your own Integration Server with this example, enter the URL of your Integration Server. Remember to use **https** at the beginning of the URL.
   - **Authentication Type**. We select HTTP Authentication. The ScienceLogic platform will pass the username and password of the current user to the Integration Server.
   - **Authentication Options**. We did not include any additional Authentication Options.

Example Using HTTP Authentication
• **Output Options.** We selected the Proxy XSL Transformation we created in the *previous section.* We selected Simple IS XSL. This XSL transformation will appear as an option when a widget is created for the proxied web service.

4. Select the [Save] button to save the new web proxy service.

### Defining a Dashboard Widget

You can use the **proxied web service widget** to view data retrieved from an external website. You can view the proxied web service widget in the **Dashboards** page ([Dashboards] tab).

To view data from our example proxied web service:

1. Go to the **Dashboards** page ([Dashboards] tab).
2. After you have configured your Dashboard and added a widget, select the **Proxied Web Service** widget type in the left NavBar of the **New Widget Configuration** page.

   **NOTE:** For details on configuring Dashboards and adding widgets, see the **Dashboards** manual.

3. In the **Widget Configuration** page, supply values in the following fields:

   - **Widget Type.** We selected Proxied Web Service.
   - **Widget Title.** We entered Integration Service.
   - **Widget Refresh.** Accept the default value.
   - **Web Service.** Select the proxied web service we defined in the *previous section.* We selected Doc IS.
- **Service Base URL.** A read-only field, populated automatically by the ScienceLogic platform when you select a value in the Web Service field. This value is the URL of the external web service that you want to display in a widget. When the platform forwards a request from the browser, the request will always start with this URL. Although the user can follow links that include this URL. The platform will not let the user make a request outside this URL.

- **Target URL.** We entered the base URL for our Integration Server. We entered https://10.100.100.16. If you would like to include an additional directory in the Service Base URL, you can specify that in this field. For example, suppose the Service Base URL displays the top-level overview page of a web service, like "www.webservice.com/intro". Suppose you always want the widget to display the sub-page "Account Details", like "www.webservice.com/intro/account". You could specify the URL of the sub-page in the Target URL field.

- **Default Transformation.** We selected the XSL transformation we defined in the previous sections and aligned with the proxied web service. We selected Simple IS.

- **Include white background behind content.** We did not select this checkbox.

4. The widget will display the following:

<table>
<thead>
<tr>
<th>Integration Service</th>
<th>Description</th>
<th>Link</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get/Update/Add/Delete User Accounts</td>
<td>/account</td>
<td></td>
</tr>
<tr>
<td>Add Alerts</td>
<td>/alert</td>
<td></td>
</tr>
<tr>
<td>Get/Update EM7 Appliances</td>
<td>/appliance</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete Asset Records</td>
<td>/asset</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete CP Theme Resources</td>
<td>/cp_theme</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add Credentials</td>
<td>/credential</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Delete Dashboards</td>
<td>/dashboard</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete Devices and Get Collected Data</td>
<td>/device</td>
<td></td>
</tr>
<tr>
<td>Get Device Classes</td>
<td>/device_class</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Groups</td>
<td>/device_group</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Templates</td>
<td>/device_template</td>
<td></td>
</tr>
<tr>
<td>Get/Update/Add/Delete Device Discovery Sessions</td>
<td>/discovery_session</td>
<td></td>
</tr>
<tr>
<td>View/Start/Stop Active Device Discovery Sessions</td>
<td>/discovery_session_active</td>
<td></td>
</tr>
<tr>
<td>Get Dynamic Application Resources</td>
<td>/dynamic_app</td>
<td></td>
</tr>
<tr>
<td>View/Update/Clear Events</td>
<td>/event</td>
<td></td>
</tr>
</tbody>
</table>

5. You can click on the links in the widget, and the widget will display the next page of data.

6. To view a page without the XSL Transformation applied, select the Options menu in the upper right of the widget and then select No Transformation.
Overview

This chapter will walk you through an example web proxy that requests data from the web interface of a Splunk server. The ScienceLogic platform then displays the top-level page of the Splunk web interface in a Dashboard widget.

The example in this chapter uses a PHP Script for authentication. The result of this example is identical to the result of the Example Using a Custom Header for Authentication. However, the Example Using a Custom Header for Authentication is easier to implement than this example, so the Example Using a Custom Header for Authentication would typically be used to create a web proxy for Splunk. This example is designed to demonstrate the features of the PHP Script Authentication method by building on the previous example.

You can use a custom PHP Script to authenticate with web services that cannot be authenticated using the HTTP Authentication or Custom Header methods. The ScienceLogic platform executes a custom PHP script to supply the required authentication information to the web service.

When you use the PHP Script method for authentication, the ScienceLogic platform constructs a PECL (PHP Extension Community Library) HTTP request and sends the request to the web service. The PECL HTTP request includes information from a standard credential and information from the custom PHP script.

To create a credential for a proxied web service using the PHP Script authentication method, you must:

- Define the credential that will provide additional information for authentication.
- Define the PHP script that will generate the authentication information.

To create this example web proxy, we will:

1. Configure Splunk to accept requests from the ScienceLogic platform.
2. Test the request.
3. Create a credential.
4. Define a DNS-based proxy.
5. Define a Proxy Web Service and a PHP Script in the ScienceLogic platform.
6. Define a Dashboard Widget to display the results of the request.

Steps 1, 2, and 4 are described in the chapter Using a Custom Header for Authentication.

Creating the Credential

Next, we must create a credential for our example. Our example uses a PHP script for authentication. To create a credential that provides the information for this type of authentication:

1. Go to the Credential Management page (System > Manage > Credentials).
2. In the Credential Management page, select the [Create] button in the upper right of the page. Select SOAP/XML Host Credential.
3. The Credential Editor page appears.
4. In the *Credential Editor* page, supply values in the following fields. If a field is not specified, the ScienceLogic platform does not use that field when authenticating a proxied web service. You can accept the default values for fields that are not specifically described in this section. The platform will use the following fields when creating the SOAP/XML credential for a proxied web service using a PHP script for authentication:

- **Profile Name.** Name of the profile. We entered *Splunk (PHP Script)*.
- **URL.** If you are creating a credential to use with a proxied web service, you can enter any valid URL in this field. The proxied web service does not use this value, but the *Credential Editor* page requires a value in this field. We entered the URL for the web interface to our Splunk server, `https://10.100.100.2:8000`.
- **HTTP Auth User.** We did not enter a value in this field. The ScienceLogic platform will authenticate using the custom header we build and send with our PHP script and the IP address of the Administration Portal or All-In-One Appliance.
- **HTTP Auth Password.** We did not enter a value in this field. The ScienceLogic platform will authenticate using the custom header we build and send with our PHP script and the IP address of the Administration Portal or All-In-One Appliance.
- **Proxy Settings.** This is an optional field. We didn't enter any values in these fields.
- **Curl Options.** This is an optional field. We didn't enter any values in this field.
- **HTTP Headers.** We did not create a custom header for this credential.

5. Select the [Save] button to save the new credential.

### Defining a Proxy Web Service and a PHP Script in the ScienceLogic platform

You can create a proxied web service in the *Proxied Web Services* page and include a PHP script to use during authentication. To create our example proxied web service:

1. Go to the *Proxied Web Services* page (Registry > Web Proxies > Proxied Web Services).
2. Select the [Create] button in the upper right of the page.
3. The *Proxy Service Editor* page appears.
4. In the **Proxy Service Editor** page, supply a value in the following fields:

- **Service Name.** Name of the proxied web service. We entered Splunk (PHP).
- **SOAP/XML Credential.** We selected the credential we created in the previous section. We selected Splunk (PHP Script).
- **Base URL.** We entered the URL for the web interface of our Splunk server (http://10.100.100.22:8000). If you are using your own Splunk server with this example, enter the URL of the web interface for your Splunk server.
- **Authentication Type.** We selected PHP script. The ScienceLogic platform will use the PHP script specified in the Authentication Script field to supply some of the authentication data.

- **Authentication Script.** We entered the following PHP script in this field:

```php
<?php
$headers['REMOTE-USER'] = $auth_user;
?>
```

- **$headers.** An associative array that the ScienceLogic platform supplies as a parameter for the setHeaders function. The setHeaders function is used to build the PECL HTTP request and allows you to define a custom header for our request. The array keys in the $headers array must be set to the name of the header. Each key must point to a value for that header. For more information about the setHeaders function, see [http://www.php.net/manual/en/function.httprequest-setheaders.php](http://www.php.net/manual/en/function.httprequest-setheaders.php)
- **REMOTE-USER.** The name of the custom header that the setHeaders function will build.
- **$auth_user.** The username of the currently logged in user. This value will be stored in the REMOTE-USER header.
• **Output Options.** We did not select a Proxy XSL Transformation.

5. Select the [Save] button to save the new web proxy service.

---

**Defining a Dashboard Widget to Display the Results**

You can use the *proxied web service widget* to view data retrieved from an external website. You can view the proxied web service widget in the **Dashboards** page ([Dashboards] tab).

To view data from our example proxied web service:

1. Go to the **Dashboards** page ([Dashboards] tab).
2. After you have configured your Dashboard and added a widget, select the **Proxied Web Service** widget type.

**NOTE:** For details on configuring Dashboards and adding widgets, see the **Dashboards** manual.

3. In the **Widget Configuration** page, supply values in the following fields:
   - **Widget Type.** We selected Proxied Web Service.
   - **Widget Title.** We entered Splunk (PHP Script).
   - **Widget Refresh.** Accept the default value.
   - **Web Service.** Select the proxied web service we defined in the *previous section*. We selected Splunk.
   - **Service Base URL.** We accepted the default value.
- **Target URL.** We entered the base URL for the web interface to our Splunk server. We entered https://10.100.100.22:8000. If you would like to include an additional directory in the **Service Base URL**, you can specify that in this field. For example, suppose the **Service Base URL** displays the top-level overview page of a web service, like "www.webservice.com/intro". Suppose you always want the widget to display the sub-page "Account Details", like "www.webservice.com/intro/account". You could specify the URL of the sub-page in the **Target URL** field.

- **Default Transformation.** We did not select an XSL transformation.

- **Include white background behind content.** We did not select this checkbox.

4. The widget will display the following:

![Welcome to Splunk](image)

5. You can click on the links in the widget, and the widget will display the next page of data.
All rights reserved.

LIMITATION OF LIABILITY AND GENERAL DISCLAIMER

ALL INFORMATION AVAILABLE IN THIS GUIDE IS PROVIDED "AS IS," WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED. SCIENCELOGIC™ AND ITS SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT.

Although ScienceLogic™ has attempted to provide accurate information on this Site, information on this Site may contain inadvertent technical inaccuracies or typographical errors, and ScienceLogic™ assumes no responsibility for the accuracy of the information. Information may be changed or updated without notice. ScienceLogic™ may also make improvements and/or changes in the products or services described in this Site at any time without notice.

Copyrights and Trademarks

ScienceLogic, the ScienceLogic logo, and EM7 are trademarks of ScienceLogic, Inc. in the United States, other countries, or both.

Below is a list of trademarks and service marks that should be credited to ScienceLogic, Inc. The ® and ™ symbols reflect the trademark registration status in the U.S. Patent and Trademark Office and may not be appropriate for materials to be distributed outside the United States.

- ScienceLogic™
- EM7™ and em7™
- Simplify IT™
- Dynamic Application™
- Relational Infrastructure Management™

The absence of a product or service name, slogan or logo from this list does not constitute a waiver of ScienceLogic’s trademark or other intellectual property rights concerning that name, slogan, or logo.

Please note that laws concerning use of trademarks or product names vary by country. Always consult a local attorney for additional guidance.

Other

If any provision of this agreement shall be unlawful, void, or for any reason unenforceable, then that provision shall be deemed severable from this agreement and shall not affect the validity and enforceability of any remaining provisions. This is the entire agreement between the parties relating to the matters contained herein.

In the U.S. and other jurisdictions, trademark owners have a duty to police the use of their marks. Therefore, if you become aware of any improper use of ScienceLogic Trademarks, including infringement or counterfeiting by third parties, report them to Science Logic’s legal department immediately. Report as much detail as possible about the misuse, including the name of the party, contact information, and copies or photographs of the potential misuse to: legal@scieneclogic.com