

Quest® Change Auditor 7.0  
**SIEM Integration Guide**



© 2018 Quest Software Inc.

**ALL RIGHTS RESERVED.**

This guide contains proprietary information protected by copyright. The software described in this guide is furnished under a software license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of the applicable agreement. No part of this guide may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording for any purpose other than the purchaser's personal use without the written permission of Quest Software Inc.

The information in this document is provided in connection with Quest Software products. No license, express or implied, by estoppel or otherwise, to any intellectual property right is granted by this document or in connection with the sale of Quest Software products. EXCEPT AS SET FORTH IN THE TERMS AND CONDITIONS AS SPECIFIED IN THE LICENSE AGREEMENT FOR THIS PRODUCT, QUEST SOFTWARE ASSUMES NO LIABILITY WHATSOEVER AND DISCLAIMS ANY EXPRESS, IMPLIED OR STATUTORY WARRANTY RELATING TO ITS PRODUCTS INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT SHALL QUEST SOFTWARE BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, PUNITIVE, SPECIAL OR INCIDENTAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF PROFITS, BUSINESS INTERRUPTION OR LOSS OF INFORMATION) ARISING OUT OF THE USE OR INABILITY TO USE THIS DOCUMENT, EVEN IF QUEST SOFTWARE HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. Quest Software makes no representations or warranties with respect to the accuracy or completeness of the contents of this document and reserves the right to make changes to specifications and product descriptions at any time without notice. Quest Software does not make any commitment to update the information contained in this document.

If you have any questions regarding your potential use of this material, contact:

Quest Software Inc.  
Attn: LEGAL Dept.  
4 Polaris Way  
Aliso Viejo, CA 92656

Refer to our website (<https://www.quest.com>) for regional and international office information.


**Patents**

Quest Software is proud of our advanced technology. Patents and pending patents may apply to this product. For the most current information about applicable patents for this product, please visit our website at <https://www.quest.com/legal>.

**Trademarks**

Quest, the Quest logo, and Join the Innovation are trademarks and registered trademarks of Quest Software Inc. For a complete list of Quest marks, visit <https://www.quest.com/legal/trademark-information.aspx>. All other trademarks and registered trademarks are property of their respective owners.

**Legend**

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

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

# Contents

<b>Integrating Change Auditor and SIEM Tools</b> .....	<b>4</b>
Webhooks in Change Auditor .....	5
Webhook terminology .....	5
Subscription configuration process .....	6
<b>Subscription Management</b> .....	<b>7</b>
Adding the PowerShell module .....	8
Viewing available commands and help .....	8
Connecting to Change Auditor .....	8
Managing subscriptions .....	9
New-CAEventWebhookSubscription .....	9
Get-CAEventWebhookSubscriptions .....	10
Set-CAEventWebhookSubscription .....	12
Remove-CAEventWebhookSubscription .....	13
Managing a Splunk integration .....	13
Working with Splunk subscriptions through the client .....	14
New-CASplunkEventSubscription .....	15
Get-CASplunkEventSubscriptions .....	16
Set-CASplunkEventSubscription .....	17
Remove-CASplunkEventSubscription .....	18
Managing an IBM QRadar integration (Preview mode) .....	19
New-CAQRadarConfiguration .....	19
New-CAQRadarEventSubscription .....	20
Get-CAQRadarEventSubscriptions .....	21
Set-CAQRadarEventSubscription .....	22
Remove-CAQRadarEventSubscription .....	23
Managing a Hewlett Packard ArcSight Logger integration (Preview mode) .....	23
Working with Change Auditor data within ArcSight .....	24
New-CAArcSightEventSubscription .....	24
Get-CAArcSightEventSubscriptions .....	25
Set-CAArcSightEventSubscription .....	27
Remove-CAArcSightEventSubscription .....	28
<b>Webhook technical insights</b> .....	<b>29</b>
Handling webhook responses .....	29
<b>About us</b> .....	<b>30</b>

---

# Integrating Change Auditor and SIEM Tools

- [Webhooks in Change Auditor](#)
- [Webhook terminology](#)
- [Subscription configuration process](#)

# Webhooks in Change Auditor

Change Auditor administrators can configure Change Auditor to send events to a third party tool using webhook technology. This technology allows you to integrate Change Auditor with Splunk, IBM QRadar, Hewlett Packard Enterprise's ArcSight, or any other tool that accepts webhook notifications.

This guide is intended for customers who want to access and reuse the rich event data gathered by Change Auditor. It describes the configuration required to implement an integration with third-party tools.

## Webhook terminology

- **Webhook receiver:** A service that has one or more webhook endpoints and is a third party tool that can receive Change Auditor events.
- **Webhook endpoint:** Specified with the NotificationUrl parameter, it is the web location where events are sent to inside the receiver.
- **Webhook subscription:** A configuration that contains information on the webhook receiver and how events should be sent. This includes a notification URL, notification interval, and the coordinator responsible for event forwarding.
- **Notification:** A message sent to the webhook receiver that contains a batch of events.

# Subscription configuration process

To begin receiving event data, you need to:

- 1 Deploy a SIEM tool that can receive and process events from Change Auditor.  
Create a webhook endpoint and configure it in your SIEM tool. The specifics for this are dependent on the tool that you are using.  
Test the webhook receiver to confirm it is working properly.
- 2 Create and configure a subscription within Change Auditor. The subscription contains information such as where to send the events, which events to include, and the coordinator responsible for event forwarding. It also contains the heartbeat notification which is a message sent to the webhook receiver that notifies it that the Change Auditor coordinator is responsive. The heartbeat notification contains the bookmark time. The bookmark is the time the last event was sent in the event notification.  
For details on creating subscriptions see [Managing a Splunk integration](#), [Managing an IBM QRadar integration \(Preview mode\)](#), and [Managing a Hewlett Packard ArcSight Logger integration \(Preview mode\)](#).
- 3 Once the subscription is created, the coordinator polls the database and continuously pushes new events to the specified notification URL in the subscription. Events are sent based on the time specified in the subscription.
- 4 Validate that events are being sent and processed by running the [Get-CAEventWebhookSubscriptions](#), [Get-CASplunkEventSubscriptions](#), [Get-CAQRadarEventSubscriptions](#), or [Get-CAArcSightEventSubscriptions](#) commands. The information in these commands indicate if the events are being received.

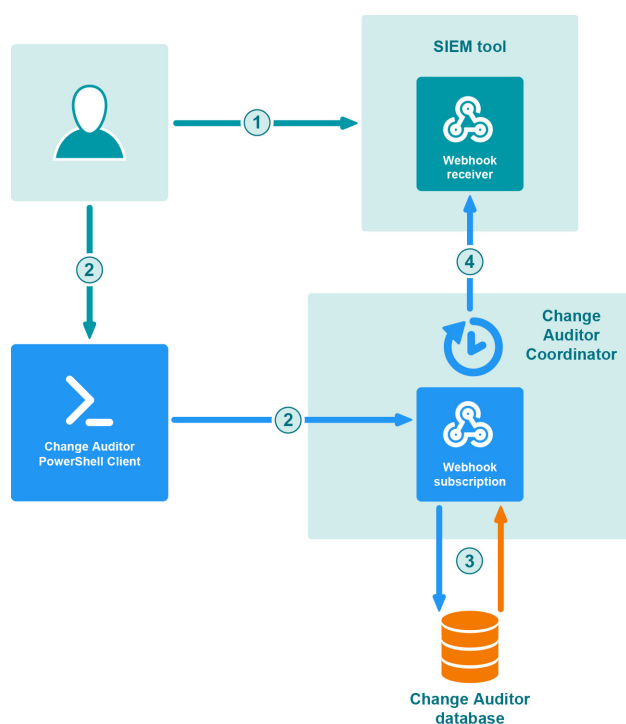


Figure 1. Webhook integration process

---

# Subscription Management

- [Adding the PowerShell module](#)
- [Viewing available commands and help](#)
- [Connecting to Change Auditor](#)
- [Managing subscriptions](#)
- [Managing a Splunk integration](#)
- [Managing an IBM QRadar integration \(Preview mode\)](#)
- [Managing a Hewlett Packard ArcSight Logger integration \(Preview mode\)](#)

# Adding the PowerShell module

Change Auditor comes with a PowerShell module for you to use to manage your environment. It is installed when you install the Windows client or a coordinator.

**i** | **NOTE:** Windows PowerShell version 3.0 or higher is required.

## To import the Change Auditor PowerShell module:

- 1 Open a Windows PowerShell window and type the following at the Windows PowerShell command prompt:

```
Import-Module <path>
```

Where "<path>" is the file path for the ChangeAuditor.PowerShell.dll assembly found in the Change Auditor Windows client or Change Auditor coordinator folder.

- 2 To ensure that the module was added, type the following at the Windows PowerShell command prompt:

```
Get-Module -All
```

The registered PowerShell modules are listed.

# Viewing available commands and help

- To view all available Change Auditor commands, enter:

```
Get-Command -Module ChangeAuditor.PowerShell
```

- To view help on each command including the syntax, enter:

```
Get-Help cmdletName
```

- To view an interactive command browser that shows you the layout of commands and the help for the commands, enter:

```
Show-Command cmdletName
```

**i** | **NOTE:** Sample scripts are available in the Change Auditor client folder. By default they are located here: C:\Program Files\Quest\ChangeAuditor\Client\PowerShell Sample Scripts

# Connecting to Change Auditor

Most Change Auditor commands require a connection to a coordinator. You can make multiple connections to different coordinators or deployments in the same script as long as the version of Change Auditor is the same.

## Example: Connect to the installation "XYZ" in the local forest

**i** | **NOTE:** This allows for fault tolerance if you have numerous coordinators by selecting the best option in the domain.

```
Connect-CAClient -InstallationName 'XYZ' -DomainName 'DomainName.com'
```



# Managing subscriptions

To begin sending event data, you need to create a subscription with Change Auditor. The subscription contains information about the URL to send the notifications and heartbeats and the event subsystems to include.

- NOTE:** You must be a member of the Change Auditor Administrators group to run these commands.
- NOTE:** These are generic commands not tied to a specific SIEM tool.

- [New-CAEventWebhookSubscription](#)
- [Get-CAEventWebhookSubscriptions](#)
- [Set-CAEventWebhookSubscription](#)
- [Remove-CAEventWebhookSubscription](#)

## New-CAEventWebhookSubscription

Use this command to create the subscription required to receive Change Auditor event data.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-NotificationUrl	Specifies where to send notifications. The notification URL is provided by the webhook receiver.
-Subsystems	Specifies an array of event subsystems from which to send events. This can be single or multiple subsystems. <b>NOTE:</b> To obtain an array of subsystems, use the Get-CAEventExportSubsystems command and filter the list to specify the required subsystems. <b>NOTE:</b> Once the subsystems are selected, they cannot be changed.
-StartTime (Optional)	Specifies date and time from which events should be sent. The default is to start sending events from the time when the subscription is created. For example: <ul style="list-style-type: none"><li>• 20 July, 2017 12:01 PM uses local time</li><li>• 2017-07-20 12:10:00Z uses UTC time</li></ul> The time will be local unless you specify the required flag to convert to UTC. <b>NOTE:</b> The time cannot be more than 30 days prior to the Change Auditor 7.0 installation date.
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. The URL is provided by the webhook receiver. <b>NOTE:</b> If no value is specified, heartbeat notifications are not sent.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the receiver. By default, this is set to 0, resulting in a continuous stream of events.

Parameter	Description
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to webhook receiver. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AuthorizationId (Optional)	Specifies the unique identifier used to confirm that the specified subscriber is authorized to accept event data. The Id is provided by webhook receiver.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events.  <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

### Example: Create a subscription that sends O365, Active Directory, and Exchange events captured since March 1 to www.quest.com

```
$startTime = Get-Date "March 1, 2018 12:00 PM"
$notificationUrl = "https://www.quest.com/api/webhook"
$selectedSubsystems = Get-CAEventExportSubsystems -Connection $connection | Where-Object DisplayName -In -Value "Office 365", "Active Directory", "Exchange"
New-CAEventWebhookSubscription -Connection $connection -NotificationUrl $notificationUrl -StartTime $startTime -Subsystems $selectedSubsystems
```

## Get-CAEventWebhookSubscriptions

Use this command to see the details of the current subscriptions.

Table 1. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-SubscriptionId (optional)	The ID of an existing webhook subscription. If specified, the command will only return the webhook subscription with that ID. If not specified, all event subscriptions are returned. You can find the SubscriptionId by running this command using just the connection information. It is also returned by the <a href="#">New-CAEventWebhookSubscription</a> command.

### Example: List defined webhook subscriptions

```
Get-CAEventWebhookSubscriptions -Connection $connection
```

#### Command output

The command returns the following information.

Table 2. Available configuration information

Setting	Description
Id	The subscription ID.
StartTime	Starting point in time for events being sent.
Subsystems	Subsystems that contain the event data being sent.
Enabled	Whether the subscription is enabled.
NotificationInterval	How often how often (in milliseconds) notifications are sent.

**Table 2. Available configuration information**

<b>Setting</b>	<b>Description</b>
HeartbeatInterval	How often (in milliseconds) heartbeat notifications are sent.
BatchSize	Batch size. (The maximum number of events to include in a single notification message.)
NotificationUrl	URL for event notifications.
HeartbeatUrl	URL for heartbeat notifications.
LastEventTime	When the last event was sent.
LastEventResponse	The last event response. Provides the response in JSON format from the event receiver.
LastHeartbeatTime	When the last heartbeat was sent.
LastHeartbeatResponse	Last heartbeat response. (For example OK, HTTP 429 - Too many events being sent., and HTTP 401 - Unauthorized access.)
EventsSent	Number of events sent.
BatchesSent	Number of batches sent.
HeartbeatsSent	Number of heartbeats sent.
BookmarkTime	Time of the event that was last sent.
AllowedCoordinators	List of coordinators permitted to send events.
LastCoordinator	The coordinator that is sending events. If the subscription is disabled, this is the last coordinator that sent events.
Internal	Whether this is an internal webhook created for a particular subscription.

# Set-CAEventWebhookSubscription

Use this command to edit the subscription.

Table 3. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAEventWebhookStatus object that corresponds to the subscription to modify. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to modify. This parameter is required if the Subscription parameter is not specified.
-NotificationUrl (Optional)	Specifies where to send notifications. The notification URL is provided by the webhook receiver.
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. The URL is provided by the webhook receiver. <b>NOTE:</b> If no value is specified, heartbeat notifications are not sent.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the receiver. By default, this is set to 0, resulting in a continuous stream of events.
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to webhook receiver. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AuthorizationId (Optional)	Specifies the unique identifier used to confirm that the specified subscriber is authorized to accept event data. The Id is provided by webhook receiver.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events. <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

## Example: Edit a webhook subscription to send events to www.quest.com for Office 365 and Active Directory

```
$subscriptionId = "ed01cc15-b67f-428d-b836-25405235dd1f"  
$notificationUrl = "https://www.quest.com/api/webhook"  
Set-CAEventWebhookSubscription -Connection $connection -SubscriptionId  
$subscriptionId -NotificationUrl $notificationUrl
```

# Remove-CAEventWebhookSubscription

Use this command to remove a subscription.

- NOTE:** You cannot use this command to remove subscriptions that are marked as internal. You can use the `Get-CAEventWebhookSubscriptions` to see which subscriptions are internal.

Table 4. Available parameters

Parameter	Description
-Connection	A connection obtained by using the <code>Connect-CAClient</code> command. See the <a href="#">Change Auditor Command Guide</a> for details.
-Subscription	The <code>PSCAEventWebhookStatus</code> object that corresponds to the subscription to remove. This parameter is required if the <code>SubscriptionId</code> parameter is not specified.
-SubscriptionId	The ID of the subscription to remove. This parameter is required if the <code>Subscription</code> parameter is not specified. Use the <a href="#">Get-CAEventWebhookSubscriptions</a> command to find the ID.

## Example: Remove a webhook subscription

```
Remove-CAEventWebhookSubscription -Connection $connection -SubscriptionId $subscriptionId
```

# Managing a Splunk integration

To begin to take advantage of the rich data gathered by Change Auditor by sending event data to Splunk, you need to create an event subscription with Change Auditor. The subscription contains information about where to send the notifications and heartbeats and the event subsystems to include.

- NOTE:** Columns that do not contain any event data will not display in Splunk.
- IMPORTANT:** To configure Splunk to receive events from Change Auditor you need to configure an HTTP event collector token in your Splunk instance.
  - Within Splunk, navigate to **Settings | Data Inputs | HTTP Event Collector**. Ensure that **All Tokens** are enabled under the Global Settings.
  - Click **New Token** and complete the steps in the wizard.
  - Copy the token. This value is required to create a Splunk subscription in Change Auditor.

Currently, you can create and manage a subscription for managed and unmanaged Splunk Cloud and Splunk Enterprise editions through the Change Auditor client or through PowerShell commands.

- [Working with Splunk subscriptions through the client](#)
- [New-CASplunkEventSubscription](#)
- [Get-CASplunkEventSubscriptions](#)
- [Set-CASplunkEventSubscription](#)
- [Remove-CASplunkEventSubscription](#)

# Working with Splunk subscriptions through the client

## **To create a Splunk subscription**

- 1 From the **Administration Tasks**, select **Configuration | Event Subscriptions**.
- 2 Click **Add** to enter the required information.
- 3 Specify where to send the event data by entering the event URL.  
For a Splunk Enterprise instance, use `https://[hostname]:[port]/services/collector/event`.  
[hostname] is the hostname of your Splunk instance, [port] is the port defined in your Splunk instance's HTTP Event Collector token page (default is 8088).  
For a Splunk Cloud instance, use:  
"https://input-[hostname]:[port]/services/collector/event".  
[hostname] is available in the address bar of an open Splunk Cloud instance and the default port is 8088.
- 4 Enter the event token.  
Splunk uses this unique identifier to confirm that the specified event URL is authorized to accept event data. The token value is created during the Splunk instance configuration.
- 5 Click **Next** to select the events to forward based on subsystem and event date. Once set, these cannot be changed by editing the subscription.
  - By default, events start sending after the subscription is created. To change when to begin sending events, click **Send events starting** and select the desired date and time. The time cannot be more than 30 days prior to the Change Auditor 7.0 installation date.
  - Select the subsystems to include in the subscription.
- 6 Click **Finish**.

## **To view existing Splunk subscription details:**

- 1 From the **Administration Tasks**, select **Configuration | Event Subscriptions**.
- 2 Expand the required subscription.  
The summary page displays the type of subscription (Target), where the events are being sent (Event URL), the subscription status (Enabled or Disabled), and when the last event was sent (Last Event).

## **To edit the event URL for a Splunk subscription**

- 1 From the **Administration Tasks**, select **Configuration | Event Subscriptions**.
- 2 Select the required subscription and click **Edit**.
- 3 Enter the new URL and click **Finish**.

## **To remove a Splunk subscription**

- 1 From the **Administration Tasks**, select **Configuration | Event Subscriptions**.
- 2 Select the required subscription and click **Delete**.
- 3 Confirm the removal.

## **To enable and disable a subscription**

- When viewing the summary information, select the status column and choose to enable or disable the subscription as required.

## To refresh the summary information

- 1 From the **Administration Tasks**, select **Configuration | Event Subscriptions**.
- 2 Click **Refresh**.

# New-CASplunkEventSubscription

Use this command to create the subscription required to send Change Auditor event data to Splunk.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-SplunkUrl	<p>Specifies the address of your Splunk instance that will receive the event data.</p> <ul style="list-style-type: none"><li>• For a Splunk Enterprise instance, use <code>https://[hostname]:[port]/services/collector/event</code> ([hostname] is the hostname of your Splunk instance, [port] is the port defined in your Splunk instance's HTTP Event Collector token page (default is 8088))</li><li>• For a Splunk cloud instance, use: <code>"https://input-[hostname]:[port]/services/collector/event"</code>. (Hostname is available in the address bar of an open Splunk cloud instance.)</li></ul> <p>For details, see the Splunk documentation on HTTP Event Collector data inputs.</p>
-EventToken	<p>The unique identifier (token) used by Splunk to confirm that the specified Splunk URL is authorized to accept event data.</p> <p>The token value is created during the Splunk instance configuration.</p> <p>For details on creating an event collector token, see the Splunk documentation on HTTP Event Collector data inputs.</p>
-Subsystems	<p>Specifies an array of event subsystems from which to send events. This can be single or multiple subsystems.</p> <p><b>NOTE:</b> To obtain an array of subsystems, use the <code>Get-CAEventExportSubsystems</code> command and filter the list to specify the required subsystems.</p> <p><b>NOTE:</b> Once the subsystems are selected, they cannot be changed.</p>
-StartTime (Optional)	<p>Specifies date and time from which events should be sent. The default is to start sending events from the time when the subscription is created.</p> <p>For example:</p> <ul style="list-style-type: none"><li>• 20 July, 2017 12:01 PM uses local time</li><li>• 2017-07-20 12:10:00Z uses UTC time</li></ul> <p>The time will be local unless you specify the required flag to convert to UTC.</p> <p><b>NOTE:</b> The time cannot be more than 30 days prior to the Change Auditor 7.0 installation date.</p>
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.

Parameter	Description
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. <b>NOTE:</b> If no value is specified, heartbeat notifications are not sent.
-HeartbeatToken (Optional)	The unique identifier (token) used by Splunk to confirm that the specified HeartbeatUrl is authorized to accept heartbeat notifications. <b>NOTE:</b> This is optional as you may have opted to send your heartbeat notifications to a URL that does not require a token for verification.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the Splunk instance. By default this is set to 0 which results in a continuous stream of events.
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatUrl. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events. <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

### Example: Create a subscription to send all subsystems event data to a Splunk instance

```
$allSubsystems = Get-CAEventExportSubsystems -Connection $connection
New-CASplunkEventSubscription -Connection $connection -SplunkUrl $splunkUrl -
EventToken $eventToken -Subsystems $allSubsystems
```

## Get-CASplunkEventSubscriptions

Use this command to see the details of the current Splunk subscriptions.

- NOTE:** The “Batches sent”, “Last event time in UTC”, “Last event response” and “Events sent” are all indicators that the events are being received by Splunk. Any failures receiving the data populate the “Last event response” property in the object with information on why the data was not received.

Table 5. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-SubscriptionId (optional)	The ID of an existing Splunk subscription. If specified, the command will only return the Splunk subscription with that ID. If not specified, all Splunk subscriptions are returned. You can find this by running this command using just the connection information. It is also returned by the <a href="#">New-CASplunkEventSubscription</a> command.

### Example: List defined Splunk subscriptions

```
Get-CASplunkEventSubscriptions -Connection $connection
```

#### Command output

The command returns the following information.

Table 6. Available configuration information

Setting	Description
Id	The subscription ID.
WebhookSubscriptionId	The webhook subscription ID.



**Table 6. Available configuration information**

Setting	Description
SplunkUrl	The URL where event data is sent.
StartTime	Starting point in time for events being sent.
Subsystems	Subsystems that contain the event data you want to send.
Enabled	Whether the subscription is enabled.
HeartbeatUrl	URL for heartbeat notifications.
LastEventTime	When the last event was sent.
LastEventResponse	Last event response. For example, statusCode = OK (200), statusCode = Bad Request (400), or statusCode = Internal Server Error (500).
LastHeartbeatTime	When the last heartbeat was sent.
LastHeartbeatResponse	Last heartbeat response. For example, statusCode = OK (200), statusCode = Bad Request (400), or statusCode = Internal Server Error (500).
EventsSent	Number of events sent.
BatchesSent	Number of batches sent.
HeartbeatsSent	Number of heartbeats sent.
NotificationInterval	How often (in milliseconds) notifications are sent.
HeartbeatInterval	How often (in milliseconds) heartbeat notificaitons are sent to the HeartbeatURL.
BatchSize	Batch size. (The maximum number of events to include in a single notification message.)
AllowedCoordinators	List of coordinators permitted to send events.
LastCoordinator	The coordinator that is sending events. If the subscription is disabled, this is the last coordinator sending the events.

## Set-CASplunkEventSubscription

Use this command to modify a Splunk subscription.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAEventWebhookStatus object that corresponds to the subscription to modify. This parameter is required if the SubscriptionId parameter is not specified. Use the <a href="#">Get-CASplunkEventSubscriptions</a> command to get a list of objects.
-SubscriptionId	The ID of the subscription to modify. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CASplunkEventSubscriptions</a> command to find the ID.

Parameter	Description
-SplunkUrl (Optional)	<p>Specifies the address of your Splunk instance that will receive the event data.</p> <ul style="list-style-type: none"> <li>For a Splunk Enterprise instance, use <code>https://[hostname]:[port]/services/collector/event</code> ([hostname] is the hostname of your Splunk instance, [port] is the port defined in your Splunk instance's HTTP Event Collector token page (default is 8088))</li> <li>For a Splunk cloud instance, use: <code>"https://input-[hostname]:[port]/services/collector/event"</code>. (Hostname is available in the address bar of an open Splunk cloud instance.)</li> </ul> <p>For details, see the Splunk documentation on HTTP Event Collector data inputs.</p>
-EventToken (Optional)	<p>The unique identifier (token) used by Splunk to confirm that the specified SplunkUri is authorized to accept event data.</p> <p>The token value is created during the Splunk instance configuration.</p> <p>For details on creating an event collector token, see the Splunk documentation on HTTP Event Collector data inputs.</p>
-BatchSize (Optional)	<p>Specifies the maximum number of events to include in a single notification. The default is 10000 events.</p>
-Enabled (Optional)	<p>Specifies whether the subscription is enabled or disabled. By default it is enabled.</p>
-HeartbeatUrl (Optional)	<p>Specifies where (URL) to send heartbeat notifications.</p> <p><b>NOTE:</b> If no value is specified, heartbeat notifications are not sent.</p>
-HeartbeatToken (Optional)	<p>The unique identifier (token) used by Splunk to confirm that the specified heartbeatUri is authorized to accept heartbeat notifications.</p> <p><b>NOTE:</b> This is optional as you may have opted to send your heartbeat notifications to a URL that does not require a token for verification.</p>
-NotificationInterval (Optional)	<p>Specifies how often (in milliseconds) notifications are sent to the Splunk instance. By default this is set to 0 which results in a continuous stream of events.</p>
-HeartbeatInterval (Optional)	<p>Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatURL. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.</p>
-AllowedCoordinators (Optional)	<p>Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events.</p> <p><b>NOTE:</b> The list order does not determine which coordinator is selected to send events.</p>

#### Example: Disable a subscription

```
Set-CASplunkEventSubscription -Connection $connection -SubscriptionId
$SubscriptionId -Enabled $false
```

## Remove-CASplunkEventSubscription

Use this command to remove a Splunk subscription.

Table 7. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAEventWebhookStatus object that corresponds to the subscription to remove. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to remove. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CASplunkEventSubscriptions</a> command to find the ID.

### Example: Remove a webhook subscription

```
Remove-CASplunkEventSubscription -Connection $connection -SubscriptionId $subscriptionId
```

## Managing an IBM QRadar integration (Preview mode)

You can take advantage of the rich data gathered by Change Auditor and use it with QRadar on-premises deployments. To begin sending event data, you need to create the QRadar configuration file and a QRadar event subscription with Change Auditor. The subscription contains information about where to send the notifications and heartbeats and the event subsystems to include.

**i** **NOTE:** The connection between Change Auditor and QRadar does not currently support TLS/SSL for secured connections. Only unsecured connections are supported for the preview release of event forwarding to on premises QRadar deployments.

- [New-CAQRadarConfiguration](#)
- [New-CAQRadarEventSubscription](#)
- [Get-CAQRadarEventSubscriptions](#)
- [Set-CAQRadarEventSubscription](#)
- [Remove-CAQRadarEventSubscription](#)

## New-CAQRadarConfiguration

Use this command to create a configuration file to import to QRadar. The file instructs QRadar on how to read and present Change Auditor events. Specifically, it defines the log source (coordinator) and maps Change Auditor event columns to QRadar event columns.

The xml file contents are returned to PowerShell standard output by default and can be piped into a file.

For information on importing a configuration file, see the QRadar documentation.

Table 2. Available parameters

Parameter	Description
-CoordinatorHostnames	Specifies a list of IPv4 addresses from which QRadar can receive events.

### Example: Create a QRadar subscription configuration file, and output it to a specified location

```
New-CAQRadarConfiguration -CoordinatorHostnames $hostnames | Out-File $filepath
```

# New-CAQRadarEventSubscription

Use this command to create the subscription required to send Change Auditor event data to QRadar.

**i** | **NOTE:** Some Change Auditor events exceed QRadar's recommended supported event data length. If a Change Auditor event exceeds this limit, the event data is continued in new QRadar events to ensure all data is stored.

**i** | **NOTE:** Columns that do not contain any event data will not display in QRadar.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-QRadarHost	Specifies the IPv4 address or FQDN (fully qualified domain name) of your QRadar instance that will receive the event data.
-Subsystems	Specifies an array of event subsystems from which to send events. This can be single or multiple subsystems.  <b>NOTE:</b> To obtain an array of subsystems, use the Get-CAEventExportSubsystems command and filter the list to specify the required subsystems.  <b>NOTE:</b> Once the subsystems are selected, they cannot be changed.
-QRadarPort (Optional)	Specifies the port number for your QRadar instance that will receive the event data. The default port is 514.
-StartTime (Optional)	Specifies date and time from which events should be sent. The default is to start sending events from the time when the subscription is created. For example: <ul style="list-style-type: none"> <li>20 July, 2017 12:01 PM uses local time</li> <li>2017-07-20 12:10:00Z uses UTC time</li> </ul> The time will be local unless you specify the required flag to convert to UTC.  <b>NOTE:</b> The time cannot be more than 30 days prior to the Change Auditor 7.0 installation date.
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. Heartbeat notifications cannot be sent directly to QRadar. To use this parameter, you must use a previously created webhook URL.  <b>NOTE:</b> If no value is specified, a heartbeat notification is not sent.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the QRadar instance. By default this is set to 0 which results in a continuous stream of events.
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatURL. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events.  <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

### Example: Create a subscription to send all subsystems event data to a QRadar instance

```
$allSubsystems = Get-CAEventExportSubsystems -Connection $connection  
New-CAQRadarEventSubscription -Connection $connection -QRadarHost $QRadarHost  
-Subsystems $allSubsystems
```

## Get-CAQRadarEventSubscriptions

Use this command to see the details of the current QRadar subscriptions.

**i** | **NOTE:** The “Batches sent”, “Last event time in UTC”, “Last event response” and “Events sent” are all indicators that the events are being received by QRadar. Any failures receiving the data populate the “Last event response” property in the object with information on why the data was not received.

Table 8. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-SubscriptionId (optional)	The ID of an existing QRadar subscription. If specified, the command will only return the QRadar subscription with that ID. If not specified, all QRadar subscriptions are returned.

### Example: List defined QRadar subscriptions

```
Get-CAQRadarEventSubscriptions -Connection $connection
```

#### Command output

The command returns the following information.

Table 9. Available configuration information

Setting	Description
Id	The subscription ID.
WebhookSubscriptionId	The webhook subscription ID.
QRadarUrl	The URL where the event data is sent.
StartTime	Starting point in time for events being sent.
Subsystems	Subsystems that contain the event data being sent.
Enabled	Whether the subscription is enabled.
HeartbeatUrl	The URL where heartbeat notifications are sent.
LastEventTime	When the last event was sent.
LastEventResponse	Last event response. For example, statusCode = OK (200), statusCode = Bad Request (400), or statusCode = Internal Server Error (500).
LastHeartbeatTime	When the last heartbeat was sent.
LastHeartbeatResponse	Last heartbeat response. For example, statusCode = OK (200), statusCode = Bad Request (400), or statusCode = Internal Server Error (500).
EventsSent	Number of events sent.
BatchesSent	Number of batches sent.
HeartbeatsSent	Number of heartbeats sent.
NotificationInterval	How often (in milliseconds) notifications are sent.
HeartbeatInterval	How often (in milliseconds) heartbeat notifications are sent.

**Table 9. Available configuration information**

Setting	Description
BatchSize	Batch size. (The maximum number of events to include in a single notification message.)
AllowedCoordinators	List of coordinators permitted to send events.
LastCoordinator	The coordinator that is sending events. If the subscription is disabled, this is the last coordinator sending the events.

## Set-CAQRadarEventSubscription

Use this command to modify a QRadar subscription.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAQRadarSubscriptionStatus object that corresponds to the subscription to modify. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to modify. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CAQRadarEventSubscriptions</a> command to find the ID.
-QRadarHost	Specifies the IPv4 address or FQDN (fully qualified domain name) of your QRadar instance that will receive the event data.
-QRadarPort (Optional)	Specifies the port number for your QRadar instance that will receive the event data. The default port is 514.
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. <b>NOTE:</b> If no value is specified, a heartbeat notification is not sent.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the QRadar instance. By default this is set to 0 which results in a continuous stream of events.
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatUrl. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events. <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

### Example: Disable a subscription

```
Set-CAQRadarEventSubscription -Connection $connection -SubscriptionId
$SubscriptionId -Enabled $false
```

# Remove-CAQRadarEventSubscription

Use this command to remove a QRadar subscription.

Table 10. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAQRadarSubscriptionStatus object that corresponds to the subscription to remove. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to remove. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CAQRadarEventSubscriptions</a> command to find the ID.

## Example: Remove a QRadar subscription

```
Remove-CAQRadarEventSubscription -Connection $connection -SubscriptionId $subscriptionId
```

# Managing a Hewlett Packard ArcSight Logger integration (Preview mode)

You can take advantage of the rich data gathered by Change Auditor and use it with ArcSight Logger. To begin sending event data, you need to create an ArcSight Logger event subscription with Change Auditor. The subscription contains information about where to send the notifications and heartbeats and the event subsystems to include.

**i** **NOTE:** The connection between Change Auditor and ArcSight Logger does not currently support TLS/SSL for secured connections. Only unsecured connections are supported for the preview release of event forwarding to ArcSight Logger.

- [Working with Change Auditor data within ArcSight](#)
- [New-CAArcSightEventSubscription](#)
- [Get-CAArcSightEventSubscriptions](#)
- [Set-CAArcSightEventSubscription](#)
- [Remove-CAArcSightEventSubscription](#)

# Working with Change Auditor data within ArcSight

The following table describes how Change Auditor event details are mapped to the event details provided in ArcSight's Common Event Format (CEF) extensions. All other Change Auditor columns not listed here will display as custom columns in ArcSight.

**NOTE:** Columns that do not contain any event data will not display in ArcSight.

Table 11. Mapping information

Change Auditor column	ArcSight column
Subsystem	deviceEventClassId
Event	name
Severity	agentSeverity
Action	categoryBehaviour
Result	categoryOutcome
Server FQDN	deviceHostName
IP Address	deviceAddress
ID	eventId
Origin IPv4	sourceAddress
Origin IPv6	c6a2
Origin	sourceHostName
User SID	sourceUserId
User	sourceUserName
Description	message
Time Detected	endTime
Time Detected	startTime

**NOTE:** Some Change Auditor events exceed ArcSight's recommended supported event data length. If a Change Auditor event exceeds this limit, the event data is continued in new ArcSight events to ensure all data is stored.

## New-CAArcSightEventSubscription

Use this command to create the subscription required to send Change Auditor event data to ArcSight.

Table 2. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-ArcSightHost	Specifies the IP address or host name of your ArcSight instance that will receive the event data.
-ArcSightPort (Optional)	Specifies the port number for your ArcSight instance that will receive the event data. The default port is 515.



Parameter	Description
-Subsystems	<p>Specifies an array of event subsystems from which to send events. This can be single or multiple subsystems.</p> <p><b>NOTE:</b> To obtain an array of subsystems, use the Get-CAEventExportSubsystems command and filter the list to specify the required subsystems.</p> <p><b>NOTE:</b> Once the subsystems are selected, they cannot be changed.</p>
-StartTime (Optional)	<p>Specifies date and time from which events should be sent. The default is to start sending events from the time when the subscription is created.</p> <p>For example:</p> <ul style="list-style-type: none"> <li>• 20 July, 2017 12:01 PM uses local time</li> <li>• 2017-07-20 12:10:00Z uses UTC time</li> </ul> <p>The time will be local unless you specify the required flag to convert to UTC.</p> <p><b>NOTE:</b> The time cannot be more than 30 days prior to the Change Auditor 7.0 installation date.</p>
-BatchSize (Optional)	<p>Specifies the maximum number of events to include in a single notification. The default is 10000 events.</p>
-Enabled (Optional)	<p>Specifies whether the subscription is enabled or disabled. By default it is enabled.</p>
-HeartbeatUrl (Optional)	<p>Specifies where (URL) to send heartbeat notifications. Heartbeat notifications cannot be sent directly to ArcSight. To use this parameter, you must use a previously created webhook URL.</p> <p><b>NOTE:</b> If no value is specified, a heartbeat notification is not sent.</p>
-NotificationInterval (Optional)	<p>Specifies how often (in milliseconds) notifications are sent to the ArcSight instance. By default this is set to 0 which results in a continuous stream of events.</p>
-HeartbeatInterval (Optional)	<p>Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatURL. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat message.</p>
-AllowedCoordinators (Optional)	<p>Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events.</p> <p><b>NOTE:</b> The list order does not determine which coordinator is selected to send events.</p>

#### Example: Create a subscription to send all subsystems event data to an ArcSight instance

```
$allSubsystems = Get-CAEventExportSubsystems -Connection $connection
New-CAArcSightEventSubscription -Connection $connection -ArcSightHost $ArcSightHost
-Subsystems $allSubsystems
```

## Get-CAArcSightEventSubscriptions

Use this command to see the details of the current ArcSight subscriptions.

**i** | **NOTE:** The “Batches sent”, “Last event time in UTC”, “Last event response” and “Events sent” are all indicators that the events are being received by ArcSight. Any failures receiving the data populate the “Last event response” property in the object with information on why the data was not received.

**Table 12. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-SubscriptionId (optional)	The ID of an existing ArcSight subscription. If specified, the command will only return the ArcSight subscription with that ID. If not specified, all ArcSight subscriptions are returned.

**Example: List defined ArcSight subscriptions**

```
Get-CAArcSightSubscriptions -Connection $connection
```

**Command output**

The command returns the following information.

**Table 13. Available configuration information**

Setting	Description
Id	The subscription ID.
WebhookSubscriptionId	The webhook subscription ID.
ArcSightHost	The IP address or host name where event data is sent.
ArcSightPort	The port where event data is sent.
StartTime	Starting point in time for events being sent.
Subsystems	Subsystems that contain the event data being sent.
Enabled	Whether the subscription is enabled.
HeartbeatUrl	URL for heartbeat notifications.
LastEventTime	When the last event was sent.
LastEventResponse	Last event response.
LastHeartbeatTime	When the last heartbeat was sent.
LastHeartbeatResponse	The last heartbeat response. For example, statusCode = OK (200), statusCode = Bad Request (400), or statusCode = Internal Server Error (500).
EventsSent	Number of events sent.
BatchesSent	Number of batches sent.
HeartbeatsSent	Number of heartbeats sent.
NotificationInterval	How often how often (in milliseconds) notifications are sent.
HeartbeatInterval	How often (in milliseconds) heartbeat notifications are sent.
BatchSize	Batch size. (The maximum number of events to include in a single notification message.)
AllowedCoordinators	List of coordinators permitted to send events.
LastCoordinator	The coordinator that is sending events. If the subscription is disabled, this is the last coordinator sending the events.

# Set-CAArcSightEventSubscription

Use this command to modify an ArcSight subscription.

**Table 2. Available parameters**

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAArcSightEventSubscriptionStatus object that corresponds to the subscription to modify. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to modify. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CAArcSightEventSubscriptions</a> command to find the ID.
-ArcSightHost (Optional)	Specifies the IP address or host name of your ArcSight instance that will receive the event data.
-ArcSightPort (Optional)	Specifies the port number for your ArcSight instance that will receive the event data. The default port is 514.
-BatchSize (Optional)	Specifies the maximum number of events to include in a single notification. The default is 10000 events.
-Enabled (Optional)	Specifies whether the subscription is enabled or disabled. By default it is enabled.
-HeartbeatUrl (Optional)	Specifies where (URL) to send heartbeat notifications. <b>NOTE:</b> If no value is specified, a heartbeat notification is not sent.
-NotificationInterval (Optional)	Specifies how often (in milliseconds) notifications are sent to the ArcSight instance. By default this is set to 0 which results in a continuous stream of events.
-HeartbeatInterval (Optional)	Specifies how often (in milliseconds) heartbeat notifications are sent to the HeartbeatUrl. By default, this is set to every 5 minutes. Setting this to 0 disables the heartbeat notifications.
-AllowedCoordinators (Optional)	Specifies the DNS or NetBIOS name of the coordinators permitted to send events. By default, any coordinator can send the events. <b>NOTE:</b> The list order does not determine which coordinator is selected to send events.

## Example: Disable a subscription

```
Set-CAArcSightEventSubscription -Connection $connection -SubscriptionId  
$SubscriptionId -Enabled $false
```

# Remove-CAArcSightEventSubscription

Use this command to remove a subscription.

Table 14. Available parameters

Parameter	Description
-Connection	A connection obtained by using the Connect-CAClient command. See the Change Auditor Command Guide for details.
-Subscription	The PSCAArcSightEventSubscriptionStatus object that corresponds to the subscription to remove. This parameter is required if the SubscriptionId parameter is not specified.
-SubscriptionId	The ID of the subscription to remove. This parameter is required if the Subscription parameter is not specified. Use the <a href="#">Get-CAArcSightEventSubscriptions</a> command to find the ID.

## Example: Remove a webhook subscription

```
Remove-CAArcSightEventSubscription -Connection $connection -SubscriptionId $subscriptionId
```

# Webhook technical insights

- [Handling webhook responses](#)

## Handling webhook responses

To see the response codes, run the associated Get command and review the LastEventResponse and LastHeartbeatResponse in the output for the following response codes:

**Table 1. Response codes**

Response code	Description
HTTP 200	Notification successfully received This response code is expected for every notification.
HTTP 429	Too many events being sent When this occurs, Change Auditor will automatically reduce the batch size when it sends its next notification.
HTTP 400	Bad Request This occurs when the receiving server is unreachable or the data is improperly formatted. Review the information provided with the response for details.
HTTP 401	Unauthorized access For example, the notification message has an incorrect or expired AuthorizationID configured in the subscription. In this case, the subscription will be disabled until the error is corrected.
HTTP 500	Internal Server Error This can be either an issue with the Change Auditor coordinator or the receiving server.

## We are more than just a name

We are on a quest to make your information technology work harder for you. That is why we build community-driven software solutions that help you spend less time on IT administration and more time on business innovation. We help you modernize your data center, get you to the cloud quicker and provide the expertise, security and accessibility you need to grow your data-driven business. Combined with Quest's invitation to the global community to be a part of its innovation, and our firm commitment to ensuring customer satisfaction, we continue to deliver solutions that have a real impact on our customers today and leave a legacy we are proud of. We are challenging the status quo by transforming into a new software company. And as your partner, we work tirelessly to make sure your information technology is designed for you and by you. This is our mission, and we are in this together. Welcome to a new Quest. You are invited to Join the Innovation™.

## Our brand, our vision. Together.

Our logo reflects our story: innovation, community and support. An important part of this story begins with the letter Q. It is a perfect circle, representing our commitment to technological precision and strength. The space in the Q itself symbolizes our need to add the missing piece—you—to the community, to the new Quest.

## Contacting Quest

For sales or other inquiries, visit [www.quest.com/contact](http://www.quest.com/contact).

## Technical support resources

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

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

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