stryker

Vocera Incoming WCTP Adapter Configuration Guide

Version 1.0.0

Notice

Stryker Corporation or its divisions or other corporate affiliated entities own, use or have applied for the following trademarks or service marks: Stryker, Vocera. All other trademarks are trademarks of their respective owners or holders. The absence of a product or service name or logo from this list does not constitute a waiver of Stryker's trademark or other intellectual property rights concerning that name or logo. Copyright © 2023 Stryker.

Contents

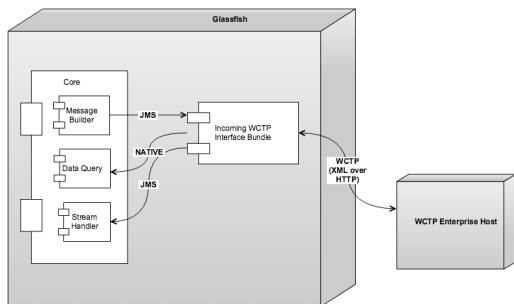# Understanding a Vocera Incoming WCTP Adapter Configuration

Configure the Vocera Incoming WCTP Adapter in the Vocera Platform Web Console to allow communication between an external system and the Vocera Platform.

Adapters send information to and receive information from the Vocera Platform, as well as monitor and collect data. Each adapter is configured to allow the Vocera Platform to communicate with a specific type of resource and any devices that the resource may control.

The Vocera Incoming WCTP Adapter implements the WCTP protocol as a server (identified as a Carrier Gateway in the specifications) with the intent of receiving inbound messages and providing responses to the senders. This adapter is designed to handle incoming messages, handle rules to send status updates including delivery errors, and also handle responses.

The adapter configuration settings are divided into two areas; Main Settings for connectivity information, and Message Types for processing incoming messages.

Viewing the Vocera Incoming WCTP Adapter Requirements

The minimum requirements for a Vocera Incoming WCTP Adapter installation are described here.

System

This Vocera Incoming WCTP Adapter depends on Vocera Platform 6.0.0 and greater.

Datasets

An adapter defines a default Dataset structure in order to function. Attributes are organized by Datasets and store the information required by the adapter. Adapters use this data during the process of receiving and sending messages.

Not all adapters require Datasets to function. When an adapter does require Datasets, the system will determine if they already exist. If they do not exist, the system will create the needed Datasets.

When creating or editing an adapter, use the following information to select the appropriate datasets in the Required Datasets section.

- The ALERT_META_DATA Dataset stores the Metadata for persisting additional Alert information.
- The RESPONSE_OPTIONS Dataset stores a response option for sending and replying with custom responses.

ALERT_META_DATA Dataset

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	message_key	N/A	True	N/A	N/A	String	Attribute that stores the locally unique message key for the message.
Attribute	allow_respons	N/A	False	N/A	False	String	Attribute that stores true if the message allows responses.
Attribute	interface	N/A	False	N/A	False	String	Attribute that stores the reference name of the processing interface.
Attribute	is_mcr	N/A	False	N/A	False	String	Attribute that stores true if the message was a multi choice response message.
Attribute	message_ident	N/A	False	N/A	False	String	Attribute that stores the external identifier for the message.
Attribute	notify_on_deli	N/A	False	N/A	False	String	Attribute that stores true if the message should notify on delivered.
Attribute	notify_on_que	N/A	False	N/A	False	String	Attribute that stores true if the message should notify on queuing.

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	notify_on_rea	N/A	False	N/A	False	String	Attribute that stores true if the message should notify on reads.
Attribute	protocol_versi	N/A	False	N/A	False	String	Attribute that stores the supported protocol version for the message.
Attribute	recipient_id	N/A	False	N/A	False	String	Attribute that stores the recipient identifier.
Attribute	sender_id	N/A	False	N/A	False	String	Attribute that stores the sender identifier.
Attribute	submit_timest	N/A	False	N/A	False	Date/Time	Attribute that stores the submit timestamp.

RESPONSE_OPTIONS Dataset

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	display_value	N/A	True	N/A	N/A	String	Attribute that stores a response value which may be used to display to the recipient.
Attribute	response_inde	N/A	True	N/A	N/A	String	Attribute that stores the index of the response in the list of responses.
Attribute	response_valu	N/A	True	N/A	N/A	String	Attribute that stores the response value to be sent back to the originating system.

Configuring a Vocera Incoming WCTP Adapter

These settings enable direct communication between the Vocera Incoming WCTP Adapter and the Vocera Platform.

Select an empty field and begin typing, or select an existing value and type over it. To keep an existing value, do not edit that field.

1. Access the Vocera Platform Web Console and navigate to the adapters.

   See Navigating the Vocera Platform Adapters on page 31 for instructions.

2. Select **New Adapter** in the Action menu, or select an adapter you wish to configure and then select **Edit**, to display the configuration fields. The configuration fields are the same for new and existing adapters.

3. Navigate to the New Adapter option, or navigate to an existing adapter to edit. See Creating a New Adapter on page 34 and Editing an Adapter on page 33 for instruction as needed.

   The configuration fields are the same for new and existing adapters.

4. Complete the configuration fields as described in the table.

Configuration Field	Description
Component Name	Click the Component Name field to display a list of the systems and devices that the Vocera Platform currently supports. Select the name of the adapter to create.
Reference Name	Enter a short descriptive name in the Reference Name field to uniquely identify an adapter instance. It may demonstrate the adapter function or other information; for example, Production adapter may differentiate a live adapter from a development or "sandbox" adapter.
Enabled	Select the Enabled checkbox to allow the Vocera Platform to use the new adapter. The Vocera Platform ignores the adapter if this option is disabled. Once an adapter is created, edit this checkbox to activate or deactivate an adapter in the system.

Configuration Field	Description
Required Datasets	If more than one dataset exists that meets the adapter's requirements, select the appropriate datasets for the new adapter to function correctly. The system searches for the datasets that meet the adapters requirements. If the datasets already exist, the system will use them. If the datasets do not exist, the system will create them automatically. Select Create in the drop-down menu to create a new dataset to meet the organization's requirements.

5. Complete the **Main Settings** configuration fields as described in the table.

   The settings described in this section provide the Vocera Platform with the explicit information required for connectivity with the inbound WCTP server. This section allows setting two values, both of which are optional and non-exportable; Sender Patterns and Security Code.

   Use the descriptions provided in this section to complete the Main Settings configuration fields.

Main Settings	Description
Sender Patterns	Enter the regular expressions to be matched to the senderID for incoming messages. If none are specified, then this configuration will handle any message which is not handled by another active configuration.
Security Code	Enter the WCTP securityCode value expected on incoming requests that support it, which are primarily messages. This value is also used on outbound messages (e.g., responses and status updates) which support it. This field is optional; however, if a security code is specified then the incoming message will require the correct security code or the message will be rejected. If a security code is not specified, then all incoming messages will be accepted.
Report Non-Matching Messages as Audit Events	Check this box to create an audit log entry when a message is received but fails to successfully match any of the specified message types.
Message Timeout	Enter the amount of time (in minutes) to wait between messages before reporting a message timeout audit event.

6. Complete the **Message Types** configuration fields as described in the table.

   At least one message type must be configured. The message types are part of the exported settings.

   Although only one message type is required to create the adapter, Implementation Specialists may add, clone, or modify message types as needed. If there are multiple message types configured, one or more can be deleted. Message types can be reordered by dragging and dropping each into the preferred order.

   See **Understading Regular Expressions** for an overview explanation and examples of Regex code and mappings.

Message Types

Alert Match, (.+) [Add] Active

Reference Name:	Alert Match
Active:	☑
Discard Message:	☐
Starting Dataset:	NurseCalls
Message Regex:	(.+)
Message Mapping:	alert_type=$1
Message Key Path:	meta_data.message_key
Sender ID Path:	
Recipient ID Path:	recipient
Priority Path:	alert_priority
Link for Responses :	response_options

Response Type Matching

Report Non-Matching Responses as Audit Events : ☑

Response Regex :	Response Link :	[Add]
.*ACCEPT.*|.*OKAY.*	accept_response	[Remove]
.*DECLINE.*|.*NO.*	decline_response	[Remove]

[Clone]

Message Type Fields	Description
Reference Name	Enter a descriptive name for the message type (e.g., for audit messages). This field is required.
Active	Select the **Active** checkbox to indicate whether or not the message type is active. Message types that are not marked Active are ignored by the adapter.
Discard Messages	Select the **Discard Messages** checkbox to not process the alarm message received if it matches this specific message type. Message types may be created for alarm messages which are expected to be received but are of no interest; by marking these for discard, the messages may be filtered from the audit log. Matched messages are treated as if they did not match any message type, with the exception of auditing.
Starting Dataset	Select the dataset from which all paths/expressions are calculated and into which the message's data should be stored. Not available for discarding message types, otherwise this field is required.
Message Regex	Enter a regular expression to match to the message text (after any decoding) of a message received from a WCTP endpoint. This is used to determine which message type matches the message. This field is required.

Message Type Fields	Description
Message Mapping	Enter a standard mapping for the message content (via the Message Regex specified) which determines how to store its information relative to the starting dataset. Not available for discarding message types, otherwise this field is required.
	Specify one or more attributes or paths, one per line, to be filled with data from the above regex. Two attribute mapping patterns are supported; plain attribute list, and statement of equality.
	Plain attribute list: Each item in the mapping is a simple attribute path. The first capture group of the matched regular expression is used as the value of the first attribute path in the list, and so on. The number of capture groups in the regex must match the number of attribute paths in the list. The syntax is dataset_link_attribute_name.attribute_name, or dataset_attribute_name.
	Statement of equality: The left-hand side of the statement is the attribute path, while the right-hand side is the value that the attribute path should be set to. On the right, use numbered captured groups (such as $1) to reference elements matched, or literal strings. The syntax is dataset_link_attribute_name.attribute_name=LITERAL, or dataset_attribute_name=$1, or example_with_two_capture_groups=$2:$1.
Message Key Path	Enter the attribute path (relative to the starting dataset) that specifies the location in which the "global identifier" for the message should be stored, so that responses and status updates can be sent. Not available for discarding message types, otherwise this field is required.
	First, configure a link attribute between the chosen Starting Dataset, and the AlertMetaData datasets. Then, in the Message Key Path field in the Incoming WCTP adapter, enter <link_name>.message_key. For example, create a link attribute named **meta_data** on the Clinicals dataset.
	Then the Message Key Path field uses the configured link attribute (meta_data) to specify the message location, **meta_data.message_key**, as shown.
Sender ID Path	Enter the location in which to store the senderID specified in the message. This field is optional.
	While the Sender ID is already stored in the 'AlertMetaData' dataset, this field allows for optionally storing the Sender ID directly off of the alert.
	The Sender ID Path field allows the system to store the sender ID in the specified attribute on the specified starting dataset. Additionally, the sender ID is also stored in the sender_id attribute of the AlertMetaData dataset regardless of what is specified in this field.
Recipient ID Path	Enter the location in which to store the recipientID specified in the message. This field is optional.
	While the Recipient ID is already stored in the 'AlertMetaData' dataset, this field allows for optionally storing the Recipient ID directly off of the alert.
	The Recipient ID Path field allows the system to store the Recipient ID in the specified attribute on the specified starting dataset. Additionally, the Recipient ID is also stored in the recipient_id attribute of the AlertMetaData dataset regardless of what is specified in this field.
Priority Path	Enter the location into which to store the priority specified in the message. This field is optional.

Message Type Fields	Description
Link For Responses	Specify the link to the ResponseOptions dataset that will be used to store all possible response options. Typing in this field will display the **Response Type Matching** configuration fields described below. Use this field when collecting all possible responses for reuse. Once these are stored, specific responses that you will be reusing can be called by the adapter. Configure a link between the Starting Dataset and the ResponseOptions datasets to store the responses. For example, create a link attribute from ResponseOptions to Alerts called **response_options**. The **response_options** link can be used in the Response Type Matching through the path entered in the **Link for Responses** field.
Response Type Matching	Enter the link to the ResponseOptions dataset where the matching responses will be stored. This field is optional. Click the **Add** button to display the configuration fields for response matching.
Report Non-Matching Responses as Audit Events	Check this box to create an audit log entry when a message contains responses but fails to successfully match to one or more of the specified response types.
Response Regex	Enter the regex to match a response type. This field is required for each Response Type added.
Response Link	Enter the name of the link on the starting dataset through which the response will be associated. This field is required for each Response Type added.
Add Message Type	Select **Add** to create additional message types.
Clone Message Type	Select **Clone** to create a duplicate of the selected message type. The reference name of the cloned message type will automatically be unique, and will be set as inactive by default.
Remove Message Type	If one or more message type is created, the ability to remove a message type becomes active. Select **Remove** to delete the message type from the adapter configuration.

7. Select one of the available options to exit the adapter configuration page. See Saving an Adapter on page 35 for details.

Understanding the Vocera Incoming WCTP Adapter Rules

Rules can be configured to trigger the Vocera Incoming WCTP Adapter to send messages to Cisco phones registered with the Cisco Unified Communications Manager.
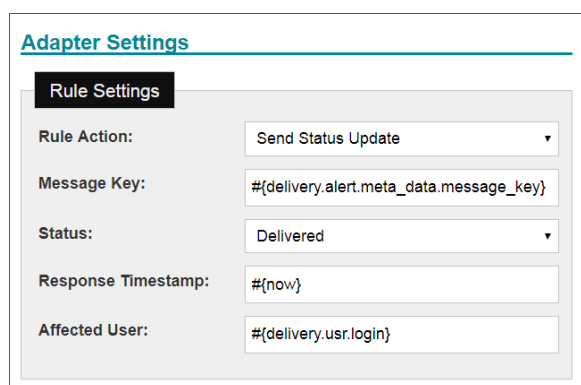
See the Vocera Platform Dataset Guide for information about working with rules. See Configuring a Vocera Incoming WCTP Adapter on page 7 for information about the adapter settings.

When triggered, the Incoming WCTP rules allow for sending status updates, message errors, and handling responses.

In the Adapter Settings, configure the Rule Settings fields to manage message delivery via a selected Rule Action. The three possible Rule Actions are displayed below, and the configuration fields are described in the table that follows.

The following image shows the fields to configure when the **Send Status Update** Rule Action is selected. This will indicate a delivery status change for the original message.

A **Send Status Update** Rule Action may be configured that will be triggered by an event that describes the state of a delivery, by configuring the rule on the DeliveryHistory dataset with configured conditions which match whether the alert is in pending or delivered status.

The following image shows the fields to configure when the **Send Delivery Error** Rule Action is selected. This will indicate that there was an error delivering the original message.

A **Send Delivery Error** Rule Action may be configured which will be triggered by a message delivery failure. Configuring the rule on the DeliveryHistory dataset with configured conditions which match if the alert failed to send or deliver for various reasons will trigger the rule action. Corresponding values for Error Code and Error Text may be specified to provide more information to the source system about the reason for the delivery failure.

Adapter Settings

Rule Settings

Rule Action:	Send Delivery Error ▾
Message Key:	#{delivery.alert.meta_data.message_key}
Error Code:	403
Error Text:	Invalid Recipient
Response Timestamp:	#{now}
Affected User:	#{delivery.usr.login}

The following image shows the fields to configure when the **Send Response** Rule Action is selected. This will indicate a response to send to the original message. Configuring the rule on the Responses dataset with configured conditions which match if the alert was accepted or declined will trigger the rule action.

The **accept_response** link attribute is configured in the Response field, in this example, to reuse the response actions configured in the Incoming WCTP adapter settings.

Adapter Settings

Rule Settings

Rule Action:	Send Response ▾
Message Key:	#{alert.meta_data.message_key}
Response:	#{alert.accept_response.response_value}
Response Timestamp:	#{now}
Responding User:	#{usr.login}

Setting	Description
Rule Action	The action to be performed when a rule is triggered. Select one option: **Send Status Update**, **Send Delivery Error**, or **Send Response**. This is a required field.
Message Key	This field will specify the global message ID of the original message. This is a required field.
Status	This field specifies the status to return (**Queued**, **Delivered**, **Read**). Note that if the original message does not request the given status, the rule will be ignored. Displays when the Rule Action named **Send Status Update** is selected.
Error Code	This field specifies the error code to send back. Displays when the Rule Action named **Send Delivery Error** is selected.
Error Text	This field specifies any optional content for the error message. Displays when the Rule Action named **Send Delivery Error** is selected.

Setting	Description
Response	This field specifies the response content, and may be blank. There is no attempt to enforce the use of one of the MCR choices, though typically the interfacing system will expect one of the options provided to be returned if choices were provided. Displays when the Rule Action named **Send Response** is selected.
Response Timestamp	This field will indicate the time of the event being sent. This field is optional. If not specified, no response timestamp is included in the message.
Affected User	This field is used to compose the sender ID used in the status updates and delivery errors. The sender ID of the response is a combination of WCTP entity-address and WCTP transport address separated by the '@' character. The transport address is the FQDN of the appliance. If specified, the sender ID will be the value specified in the field as the entity-address portion. If the Affected User is not specified, then the value of the sender ID in the response will be just the transport address. For example: "bob@vocera.engage.com" when specified, or just "vocera.engage.com" when not specified. Displays when the Rule Action named **Send Status Update** or **Send Delivery Error** is selected.
Responding User	This field is used to compose the sender ID used in the response. The sender ID of the response is a combination of WCTP entity-address and WCTP transport address separated by the '@' character. The transport address is the FQDN of the appliance. If specified, the sender ID will be the value specified in the field as the entity-address portion. If the Responding User is not specified, then the value of the sender ID in the response will be just the transport address. For example: "bob@vocera.engage.com" when specified, or just "vocera.engage.com" when not specified. Displays when the Rule Action named **Send Response** is selected.

Integrating Incoming WCTP with Vocera Platform

An Vocera Incoming WCTP Adapter integration requires storing the originating system alert in Vocera Platform datasets for use in rules and workflows.

The Vocera Incoming WCTP Adapter has the unique ability to process alerts and store responses dynamically. These responses can be used later in rules to send messages with response options.

The multiple choice response (MCR) matching behavior is controlled by the alerts and responses stored in Vocera datasets. See **Mulitple Choice Response (MCR) Matching** for information about how an external alerting system's multiple arbitrary responses can be handled.

Configuring AlertMetaData Dataset

The AlertMetaData dataset stores all additional alerting options that should be persisted across a failover.

The Vocera Incoming WCTP Adapter must store additional information about alerts in the standard Vocera data model, because of the nature of the response mechanism and the need to persist data across a failover. This data will be persisted in a separate dataset, the AlertMetaData dataset.

The AlertMetaData dataset is tied to the original alert by a locally unique identifier, referred to as the Message Key, which is stored on the dataset as the message_key attribute.

The Vocera Incoming WCTP Adapter is configured with a message type for matching incoming messages; the starting dataset is selected when creating that message type for matching.

ALERT_META_DATA Dataset Attributes

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	message_key	N/A	True	N/A	N/A	String	Attribute that stores the locally unique message key for the message. This key is generated by the adapter, it uniquely identifies the message, and is the key for the AlertMetaData dataset.
Attribute	allow_respons	N/A	False	N/A	False	String	Attribute that stores true if the message allows responses.
Attribute	interface	N/A	False	N/A	False	String	Attribute that stores the reference name (ID) of the interface which was used to process the message.
Attribute	is_mcr	N/A	False	N/A	False	String	Attribute that stores true if the message was a multi choice response (MCR) message.
Attribute	message_ident	N/A	False	N/A	False	String	Attribute that stores the message identifier supplied by the source system.
Attribute	notify_on_deli	N/A	False	N/A	False	String	Attribute that stores true if status updates can be sent for delivered messages.

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	notify_on_que	N/A	False	N/A	False	String	Attribute that stores true if status updates can be sent for queued messages.
Attribute	notify_on_rea	N/A	False	N/A	False	String	Attribute that stores true if status updates can be sent for read messages.
Attribute	protocol_versi	N/A	False	N/A	False	String	Attribute that stores the supported protocol version used by the inbound alertmessage.
Attribute	recipient_id	N/A	False	N/A	False	String	Attribute that stores the message's specified recipient identifier.
Attribute	sender_id	N/A	False	N/A	False	String	Attribute that stores the identity of the sender.
Attribute	submit_timest	N/A	False	N/A	False	Date/Time	Attribute that stores the time that the incoming WCTP message was submitted.

Configuring the AlertMetaData Dataset

First create a one-to-one link between either the Alerts dataset or other Starting dataset, and the AlertMetaData dataset. Then configure the Vocera Incoming WCTP Adapter, specifying this link when providing the message_key attribute in the Message Type settings.

Create Links on the Starting Dataset to Connect with the AlertMetaData Dataset

Create the link attribute on the starting dataset that is specified in the rule on the dataset.

Name	Reverse Name	Linked Dataset	Type	Key	Reverse Key	index
meta_data	alerts	AlertMetaData	one-to-one	false	false	false

Specify the Message Key and Link in the Adapter Configuration

Use the link attribute that connects the starting dataset to the AlertMetaData dataset (meta_data) and join it with the message_key attribute on the AlertMetaData dataset to configure the Message Key Path field in the Message Type settings in the Vocera Incoming WCTP Adapter.

Configuring the ResponseOptions Dataset

Because an MCR message may contain multiple responses, the Vocera Incoming WCTP Adapter must be able to store each response type and have a default Accept response and a default Decline response.

Each MCR type may vary in response options between adapter configurations and message types, so each message type should be configured with a regex to match the correct Accept response, and a separate regex to match the correct Decline response. The message type that the message was matched to is configured with the Starting Dataset.

RESPONSE_OPTIONS Dataset Attributes

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	display_value	N/A	True	N/A	N/A	String	Attribute that stores a response value which may be used to display to the recipient. This is a optional separate "display value" to be shown to the user when choosing the response.
Attribute	response_inde	N/A	True	N/A	N/A	String	Attribute that stores the index of the response in the list of responses. This is the numerical order in which the responses were received, or some order determined by the adapter.

Element	Name	Reverse Name	Key	Reverse Key	Required	Type	Description
Attribute	response_valu	N/A	True	N/A	N/A	String	Attribute that stores the response value to be sent back to the originating system.

Configuring the ResponseOptions Dataset

First create a many-to-many link between either the Alerts dataset or other Starting Dataset, and the ResponseOptions dataset. Then configure the Vocera Incoming WCTP Adapter, specifying this link when providing the Link for Responses field in the Message Type settings. Finally, create many-to-one links between the Starting Dataset and the ResponseOptions dataset so that the specific response can be accessed in a rule.

Create Links on the Starting Dataset to Connect with the ResponseOptions Dataset

Create the link attribute on the starting dataset that is specified in the rule on the dataset. Create the response_options attribute to link to the ResponseOptions dataset from the Alerts dataset in this example.

Name	Reverse Name	Linked Dataset	Type	Key	Reverse Key	index
response_option	alerts	ResponseOption	many-to-many	false	false	false

Specify the Link in ResponseOptions

Use the link attribute created to connect with the ResponseOptions dataset in the Vocera Incoming WCTP Adapter configuration. Provide the new link attribute in the Link for Responses field in the Message Type settings.

Create Links on Starting Dataset for Sending Rules

Create many-to-one links between the StartingDataset dataset and the ResponseOptions when matching message types. These links provide the specific response to be accessed in a rule to send to the user or to send back to the originating system.

Name	Reverse Name	Linked Dataset	Type	Key	Reverse Key	Index
accept_response	accepted_alert	ResponseOption	many-to-one	false	false	false
decline_respons	declined_alert	ResponseOption	many-to-one	false	false	false

Specify the Links in the Adapter Configuration

Use the link attributes in the Response Type Matching settings in the Vocera Incoming WCTP Adapter configuration.

Use the Links in a Send Rule on a Dataset

These links can then be supplied in a rule to send a message to a user or to send back to the originating system. Configure the Response List field in the Rule Settings on a dataset rule that uses the Vocera adapter in the example below.

Understanding Integration through Examples

Use the following examples to implement the Incoming WCTP features.

The examples that follow illustrate the use of only a single trigger condition for determining the appropriate rule action. In practice, additional conditions would likely be necessary to ensure that only responses, status updates, and errors related to messages received through the Vocera Incoming WCTP Adapterr will trigger these rules. The recommended additional conditions are as follows:

- The correct condition for sending the given response exists. If the response is being triggered by an end user response, this may be a "Response Is Accept" or "Response Is Decline" condition on the Responses dataset.
- The appropriate response link (alert.accept_response or alert.decline_response) is non-null.
- The appropriate meta_data link (alert.meta_data.message_key) is non-null.
- Optionally, for instance if there are multiple multi-response adapter instances, there might need to be a condition to check the interface reference name stored in the metadata (alert.meta_data.interface), or some other mechanism to fire the rule for the correct adapter instance.

Trigger Conditions			Add Condition
A message key exists for the alert	▼	alert.meta_data not null	Remove
Response is a decline	▼	action starts with Decline	Remove
The alert has a decline response	▼	alert.decline_response not null	Remove
The alert is for the IncomingWCTP Interface	▼	alert.meta_data.interface equal to IncomingWCTP	Remove

Sending an Accept Response

When an alert has been accepted by the recipient in one of the Vocera-supported endpoints, we can send back a corresponding response to the originating system using an Incoming WCTP Send Response Rule.

The rule should be configured on the Responses dataset with a condition which matches the affirmative response (i.e., action starts with 'accept'). The rule can then be configured to send back one of the values that was stored in the the Response Type matching section of the adapter configuration using the accept_response link.

Please note that we are sending back the response_value attribute of the ResponseOptions dataset, as opposed to the display_value, since this is the value that the source system will generally be expecting.

Sending a Decline Response

When an alert has been declined by the recipient in one of the Vocera-supported endpoints, we can send back a corresponding response to the originating system using an Incoming WCTP Send Response Rule.

The rule should be configured on the Responses dataset with a condition which matches the negative response (i.e., action starts with 'Decline'). The rule can then be configured to send back one of the values that was stored in the the Response Type matching section of the adapter configuration using the decline_response link.

Please note that we are sending back the response_value attribute of the ResponseOptions dataset, as opposed to the display_value, since this is the value that the source system will generally be expecting.

Edit Send a Declined response to WCTP Source System

Purpose:	Send a Declined response to WCTP Source System
Adapter:	**IncomingWCTP**
Defer Delivery By:	0
Don't send back to originating adapter:	☐
Active?:	☑

Trigger Events

Create Event:	☑
Update Event:	☐

Trigger Conditions   Add Condition

Response is a decline ▼   action starts with Decline   Remove

Adapter Settings

Rule Settings

Rule Action:	Send Response ▼
Message Key:	#{alert.meta_data.message_key}
Response:	#{alert.decline_response.response_value}
Response Timestamp:	#{now}
Responding User:	#{usr.login}

Save   Reset   Cancel

Sending a Queued Status Update

When an alert has been queued for delivery to one of the Vocera-supported endpoints, we can send back a status update to notify the originating system that we have successfully received the alert and are attempting to send the alert out to the recipient.

This can be accomplished by configuring an Incoming WCTP Send Status Update Rule on the DeliveryHistory dataset with a condition which matches when an alert is queued for delivery (i.e., status equal to 'Queued') and selecting the 'Queued' status in the dropdown for the Send Status Update Rule.

Datasets › DeliveryHistory › Rules › Send a Queued Status Update › Edit

Edit Send a Queued Status Update

Purpose: Send a Queued Status Update

Adapter: **IncomingWCTP**

Defer Delivery By: 0

Don't send back to originating adapter: ☐

Active?: ☑

Trigger Events

Create Event: ☑

Update Event: ☐

Trigger Conditions   Add Condition

Message was queued ▼   status equal to Queued   Remove

Adapter Settings

Rule Settings

Rule Action: Send Status Update ▼

Message Key: #{alert.meta_data.message_key}

Status: Queued ▼

Response Timestamp: #{now}

Affected User: #{delivery.usr.login}

Save   Reset   Cancel

Sending a Successfully Delivered Status Update

When an alert has been successfully delivered to one of the Vocera-supported endpoints, we can send back a status update to notify the originating system that we have successfully received the alert and have delivered it to the recipient.

This can be accomplished by configuring an Incoming WCTP Send Status Update Rule on the DeliveryHistory dataset with a condition which matches when an alert is queued for delivery (i.e., status equal to 'Delivered') and selecting the 'Delivered' status in the dropdown for the Send Status Update Rule.

Datasets 〉 DeliveryHistory 〉 Rules 〉 Send a Delivered Status Update 〉 Edit

Edit Send a Delivered Status Update

Purpose:	Send a Delivered Status Update
Adapter:	**IncomingWCTP**
Defer Delivery By:	0
Don't send back to originating adapter:	☐
Active?:	☑

Trigger Events

Create Event:	☑
Update Event:	☐

Trigger Conditions   Add Condition

Message is delivered	▾	status equal to Delivered	Remove

Adapter Settings

Rule Settings

Rule Action:	Send Status Update ▾
Message Key:	#{alert.meta_data.message_key}
Status:	Delivered ▾
Response Timestamp:	#{now}
Affected User:	#{delivery.usr.login}

Save   Reset   Cancel

Sending a Delivery Failure Status Update

When an alert has failed to deliver to one of the Vocera-supported endpoints, we can send back a status update to notify the originating system that we have successfully received the alert and have failed to deliver it to the recipient.

Here we have the option to send back a custom error code and error message. This can be accomplished by configuring an Incoming WCTP Send Delivery Error Rule on the DeliveryHistory dataset with a condition which matches when an alert fails to deliver (i.e., status equal to 'Error') and providing an error code and error message.

If there are several different error codes that will be useful for the originating system, more conditions could be conceived to determine the reason for the failed delivery (likely using the link DeliveryHistory.delivery.status_text), and using an error code and error message which correspond with the reason for failure.

Datasets 〉 DeliveryHistory 〉 Rules 〉 Send a Delivery Failure 〉 Edit

Edit Send a Delivery Failure

Purpose:	Send a Delivery Failure
Adapter:	**IncomingWCTP**
Defer Delivery By:	0
Don't send back to originating adapter:	☐
Active?:	☑

Trigger Events

Create Event:	☑
Update Event:	☐

Trigger Conditions   Add Condition

Message failed to deliver ▾	status equal to Error	Remove

Adapter Settings

Rule Settings

Rule Action:	Send Delivery Error ▾
Message Key:	#{alert.meta_data.message_key}
Error Code:	403
Error Text:	Invalid Recipient
Response Timestamp:	#{now}
Affected User:	#{delivery.usr.login}

Save   Reset   Cancel

Understanding Adapter Installation

Adapters are installed on the Vocera Platform in a solution package, or individually as needed by the customer.

The Vocera Platform uses adapters to integrate with external systems and devices. Each adapter is configured by the user to include information that will allow the Vocera Platform to communicate and interact with a specific type of resource and, depending on the adapter, devices that resource may control. Adapters can allow the Vocera Platform to monitor and collect data, as well as send data out, when triggered manually or automatically.

When implementing Vocera Platform at a customer site, use this document to install an adapter that is not supplied in the Gold Image. Otherwise, you will install a needed adapter when instructed in the solution package installation process described in the Vocera Platform Installation Guide.

Recreating a Repository

In the event that the repository reference file has been compromised, you can re-create the platform repository.

This information should be specified on the related adapter's Release Information page in the wiki. See **Releases** and navigate to the needed adapter.

1. Verify that the adapter resides in a repository which is in '**/etc/yum.repos.d/**'.
2. If the **repolist** or **yum** commands fail, verify that the file exists and try again. For example, use the following code to verify the repository exists on the Vocera Platform appliance:

```
[tpx-admin@engage log]$ cat /etc/yum.repos.d/vocera.repo
```

3. Verify the output appears as shown.

```
#--------------------------------------------------------------------------------
# NOTICE: Only use the General Availability (platform-6.X-ga) repository for customer
 deployments.
# Use of Controlled Release (platform-6.X-cr) or Software Quality Assurance
 (platform-6.X-sqa) in
# accordance to process QOP-75-01 Production Work Order and History Record, contact
 your
# manager for questions.
#--------------------------------------------------------------------------------
[Platform-6.0]
name=Platform-6.0
baseurl=https://box.voceracommunications.com/Platform-6.0-GA
enabled=1
gpgcheck=0
```

Installing an Adapter

Install or uninstall a Vocera Platform adapter at a customer site on a Vocera system for a customer.

Execute the following steps using the system's command prompt.

1. Verify that the adapter resides in a repository which is in '**/etc/yum.repos.d/**'.
2. Run the following commands:

```
sudo yum clean all
sudo yum check-updates
```

3. Verify that the rpm package to be installed is available using the following command:

```
sudo yum list available | grep extension
```

4. Install the adapter by specifying its rpm package name in place of <package-name> in the code below. (This information should be specified on the related Release Information page in the wiki; see **Release Notes**.)

```
sudo yum install <package-name>
```

5. Uninstall an adapter by specifying its rpm package name in place of <package-name> in the code below. (This information should be specified on the related Release Notes page; see **Release Notes**.)

```
sudo yum remove <package name>
```

Practicing an Adapter Installation

Replicate these steps using the needed adapter package, in order to install adapters other than the example given here.

1. Verify the repo file contains the repos up to and including the release of interest.

```
[tpx-admin@engage log]$ cat /etc/yum.repos.d/vocera.repo
#---------------------------------------------------------------------------
# NOTICE: Only use the General Availability (platform-6.X-ga) repository for customer
 deployments.
# Use of Controlled Release (platform-6.X-cr) or Software Quality Assurance
 (platform-6.X-sqa) in
# accordance to process QOP-75-01 Production Work Order and History Record, contact
 your
# manager for questions.
#---------------------------------------------------------------------------
[Platform-6.0]
name=Platform-6.0
baseurl=https://box.voceracommunications.com/Platform-6.0-GA
enabled=1
gpgcheck=0
```

2. Execute the following commands:

```
[tpx-admin@engage log] $ sudo yum check-updates
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
 subscription-manager to register.
Quartz
                                               | 3.6 kB  00:00:00

(1/2): Quartz/group_gz
                                               |  483 B  00:00:00

(2/2): Quartz/primary_db
                                               |  29 kB  00:00:00
```

3. Verify the package is available, using the following command:

```
[tpx-admin@engage log] $ sudo yum list available | grep extension
   extension-navicare-interface.x86_64                    1.3.6-0              Platform 5.0
```

4. Install the needed adapter; in this example, install the Navicare adapter:

```
[tpx-admin@engage log] $ sudo yum install extension-navicare-interface
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
 subscription-manager to register.
Resolving Dependencies
--> Running transaction check
---> Package extension-navicare-interface.x86_64 0:1.3.6-0 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
 Package                            Arch
 Version                   Repository                   Size
================================================================================
Installing:
 extension-navicare-interface                      x86_64
 1.3.3-0                   Quartz                       59 k

Transaction Summary
================================================================================
Install  1 Package

Total download size: 59 k
Installed size: 62 k
Is this ok [y/d/N]: y
Downloading packages:
extension-navicare-interface-1.3.6-0.x86_64.rpm
                                          |  59 kB   00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : extension-navicare-interface-1.3.6-0.x86_64
                                                      1/1
  Verifying  : extension-navicare-interface-1.3.6-0.x86_64
                                                      1/1


Installed:
  extension-navicare-interface.x86_64 0:1.3.6-0


  Complete!
```

5. This completes the steps to install an adapter.

Navigating the Vocera Platform Adapters

Access the Adapters tab and use the filter or search tools to display a specific adapter.
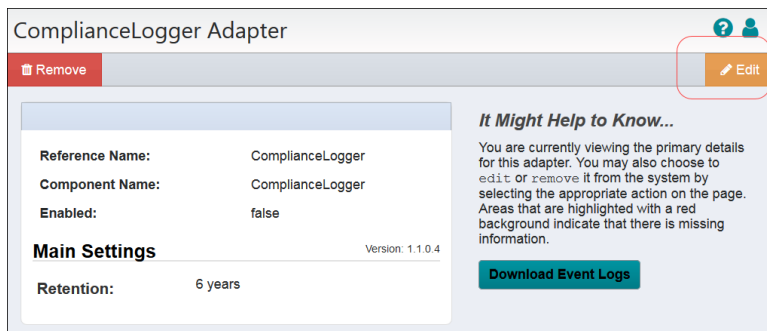
This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

1. Access the Vocera Platform Web Console and sign in with your system credentials.
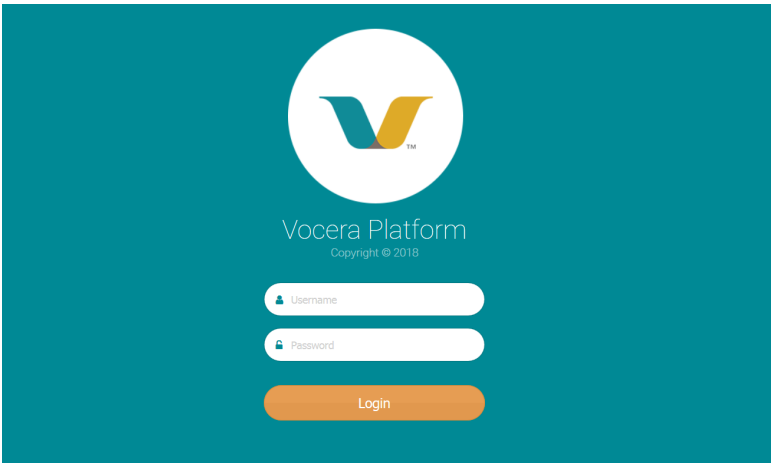
2. Select **Settings > Adapters** in the navigation menu.
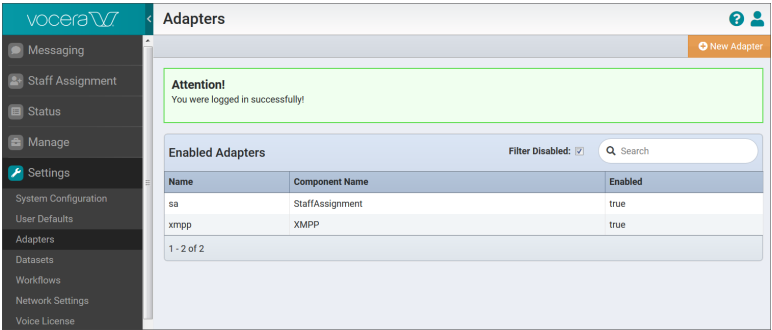
   The **Adapters** page displays.

3. Select an adapter to work with from the list displayed in the grid, or select the New Adapter Action option to create a new adapter.

   On the **Adapters** page you can identify adapters by their name or component name. The Enabled column (displaying a true or false status) indicates whether the adapter is active on the system, or disabled.

   The bottom row of the grid reports the number of adapters displayed, of the available adapters.

   The Filter Disabled box is checked by default, and displays only the enabled adapters that are configured on the Vocera Platform.
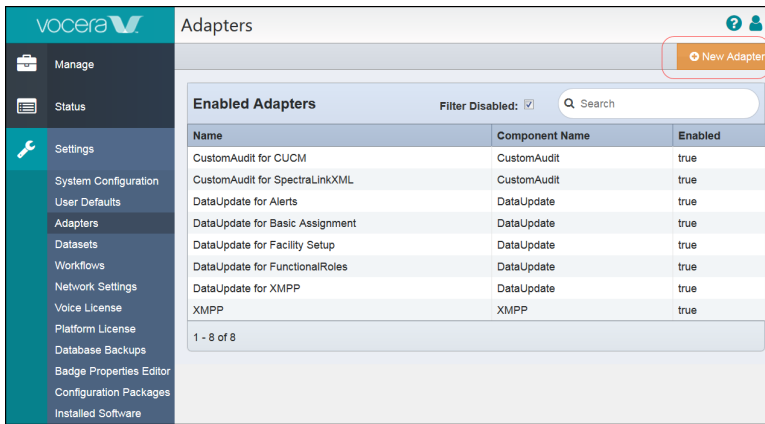
4. Uncheck the **Filter Disabled** box to display all the adapters that have been installed, including those that are not currently enabled. The column title now displays **All Adapters**.

The Filter Disabled box is checked by default.
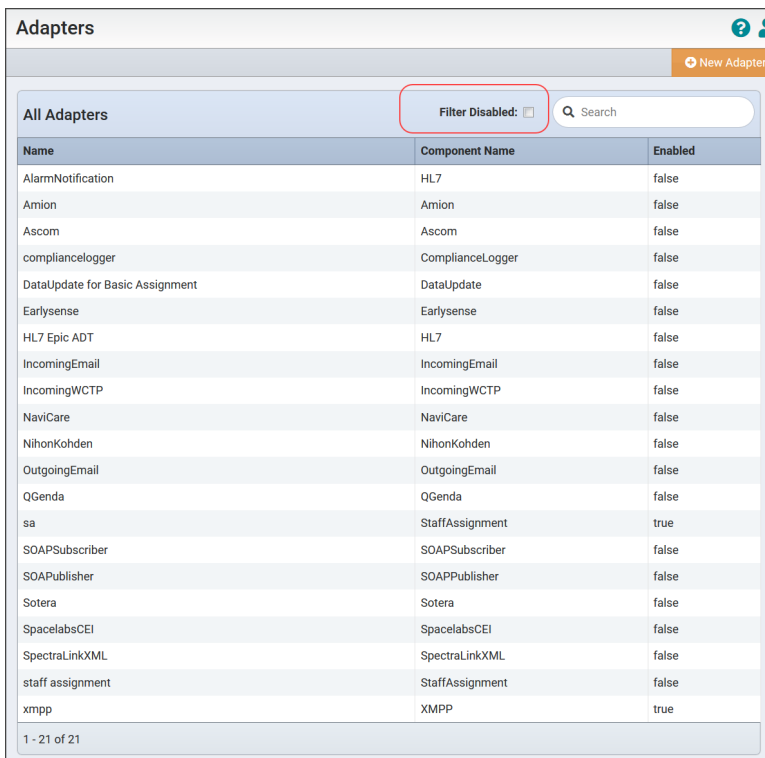
5. Enter a term in the **Search** field to locate a needed adapter on the system.

The search field is identified by a text field with a magnifying glass icon. The search is performed on the Name and Component Name columns.

When results are returned, the column header displays **Adapters Search Results** and an **x** icon allows you to clear the search field.
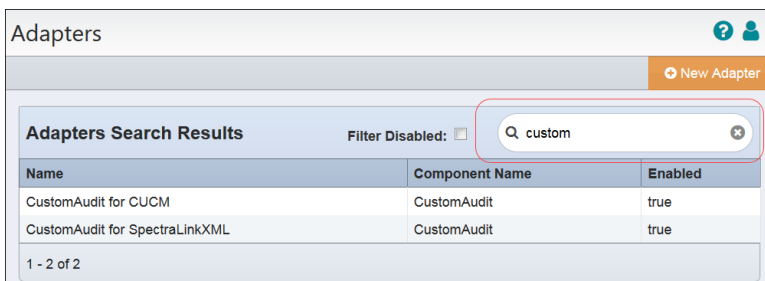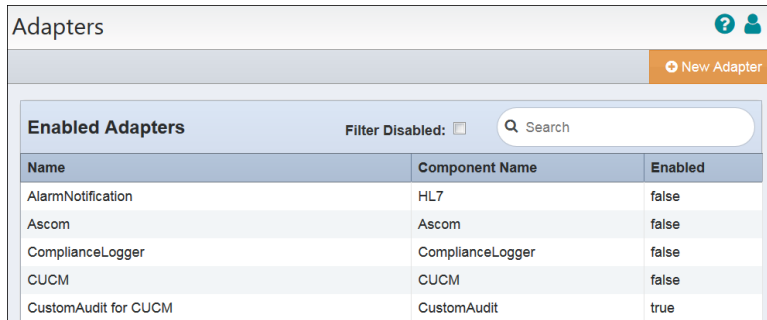
Editing an Adapter

Edit an adapter that has been installed on the Vocera Platform.

This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

1. Access the Vocera Platform Web Console and navigate to the adapters.

   See Navigating the Vocera Platform Adapters on page 31 for instructions.

2. Select the adapter to edit in the **Adapters** list.

Name	Component Name	Enabled
AlarmNotification	HL7	false
Ascom	Ascom	false
ComplianceLogger	ComplianceLogger	false
CUCM	CUCM	false
CustomAudit for CUCM	CustomAudit	true

3. Select **Edit** in the adapter's menu.
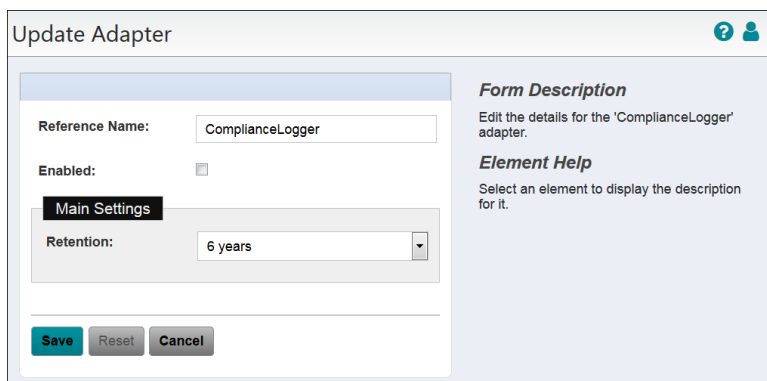
   The **Update Adapter** page for the adapter displays.

4. Edit the adapter's settings to revise the configuration as needed. See the adapter-specific configuration page for details on working with settings for this adapter.

   Select an empty field and begin typing, or select an existing value and type over it. To keep an existing value, do not edit that field.

5. Select one of the options to exit the **Update Adapter** page. See Saving an Adapter on page 35 for details.

Creating a New Adapter

Access the Vocera Platform Web Console to work with adapters, or create a new adapter when prompted in the package import process.

This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

1. Access the Vocera Platform Web Console and navigate to the adapters.

   See Navigating the Vocera Platform Adapters on page 31 for instructions.
2. Select **New Adapter** in the Action menu on the Adapters page.
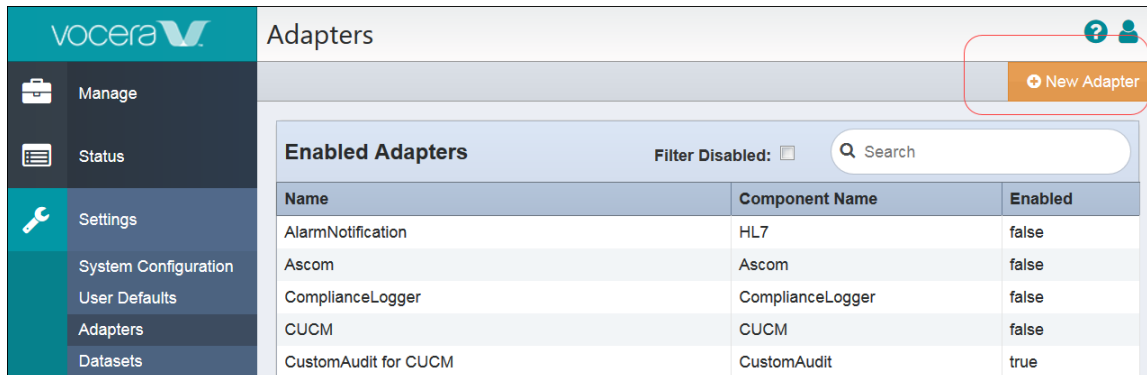
   The **Create a New Adapter** dialog displays.
3. Complete the configuration fields.

Name	Description
Component Name *	Select the Component Name field dropdown arrow to display a list of the systems and devices that Vocera currently supports. Select the name of the adapter to create.
Reference Name	Enter a short descriptive name in the Reference Name field to uniquely identify an adapter instance. It may demonstrate the adapter function or other information; for example, Production adapter may differentiate a live adapter from a development or "sandbox" adapter.
Enabled	Select the Enabled check box to allow Vocera Platform to use the new adapter. Vocera ignores the adapter if this option is disabled.

4. Select **Upload Bundle** in the Action menu to install a package on a Vocera Platform.

   Use the Upload Bundle feature to install when the adapter is not available in the Component Name dropdown list, and you have downloaded the needed adapter bundle to a storage location.
5. Click on **Browse** to navigate to the bundle to install.

6. Select one of the Action options to exit from the Upload a Bundle dialog.
   - **Upload**: Upload the selected bundle to the appliance.
   - **Cancel**: Close the Upload a Bundle dialog without making a change to the system.

Saving an Adapter

Close an adapter configuration dialog using the Save, Reset, or Cancel options.

This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

When creating a new adapter, the options at the bottom of the adapter configuration page are Save, and Cancel.

When editing an existing adapter, the options are Save, Reset, and Cancel.

Choose an option to close the dialog:

Option	Description
Save	Select Save to store the adapter configuration in the system, when the fields are set to desired specifications.
Cancel	Select Cancel to close the configuration window without saving your changes to the system.
Reset	Select Reset to clear all fields without closing the window, in order to select other specifications for the adapter's settings.

Deactivating an Adapter

Temporarily deactivate an adapter to avoid unintentional use of it in an implementation.

This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

1. Access the Vocera Platform Web Console and navigate to the adapter to deactivate.

   See Navigating the Vocera Platform Adapters on page 31 for instructions.

2. Select **Edit** in the Actions menu to access the Update page for the adapter.

3. Un-check the **Enabled** box to temporarily deactivate the adapter.

   When deactivated, the Vocera system will ignore the adapter. You can easily enable or disable the adapter at any time.

4. Select one of the options to exit the **Update Adapter** page. See Saving an Adapter on page 35 for details.

Removing an Adapter

Permanently remove an adapter from the Vocera system.

This page is used by all the adapter guides, and therefore, the adapter used as an example here may not be the adapter that you are working with currently.

Use the remove function to permanently delete the adapter from the system. Alternatively, you can disable an adapter and the Vocera system will ignore it.

⚠️ **Warning:** Remove cannot be undone. If any system features use this adapter, removing the adapter prevents the features from functioning.
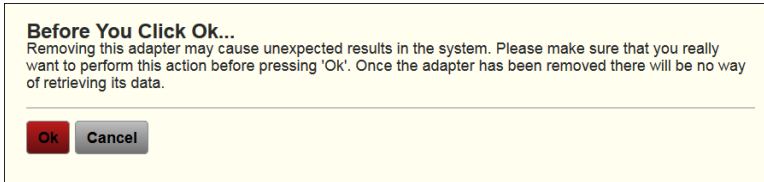
1. Access the Vocera Platform Web Console and navigate to the adapter to remove.

   See Navigating the Vocera Platform Adapters on page 31 for instructions.

2. Select **Remove** in the Actions menu to permanently delete the adapter.

NAVIGATING THE VOCERA PLATFORM ADAPTERS

3. Click **Ok** in the confirmation window.

Before You Click Ok...
Removing this adapter may cause unexpected results in the system. Please make sure that you really want to perform this action before pressing 'Ok'. Once the adapter has been removed there will be no way of retrieving its data.

[Ok] [Cancel]

- **Ok**: Confirm the choice to remove the adapter from the system.
- **Cancel**: Return to the adapter page without making a change.

4. Confirm that the adapter no longer displays in the Adapters list view, when a success message displays.

Adapters

⊕ New Adapter

Attention!
The adapter was removed from the system successfully.