

# IBM Resilient



## Incident Response Platform Integrations

### Slack Function V1.0.2

Release Date: August 2020

Resilient Functions simplify development of integrations by wrapping each activity into an individual workflow component. These components can be easily installed, then used and combined in Resilient workflows. The Resilient platform sends data to the function component that performs an activity, and returns the results to the workflow. The results can then be used by scripts, rules, and workflow decision points to dynamically orchestrate the security incident response activities.

This guide describes the Slack Integration.

## Overview

Slack is an online communications solution allowing communities to communicate as groups or directly with each other through conversations and video conferences in Slack channels. This Resilient platform functions-based integration enables Incident, Note, Artifact, Task, and Attachment data to be shared in Slack. Users can create or designate private or public channels in a Slack workspace and invite Slack users to the channels. Users can also customize data from the Resilient objects to post in a channel, as well as archive channels.

There are three functions and eight example workflows in this integration package. This document describes the included functions, how to configure them in custom workflows, and demonstrates additional customization options.

## History

Version	Comment
1.0.2	Support for App Host, Proxy support added

## App Host Installation

All the components for running Slack in a container already exist when using the App Host app. The remainder of this section details the Slack configuration file changes.

It's possible to override the template used for archiving a channel.

Use the app.config setting: `template_file=/var/rescircuits/slack_template.jinja2` to reference the template file named `slack_template.jinja2` at location `/var/rescircuits`. See the Template file section below for the default jinaj2 code.

## Integration Server Installation

Before installing, verify that your environment meets the following prerequisites:

- Resilient platform is version 35 or later.
- You have a Resilient account to use for the integrations. This can be any account that has the permission to view and modify administrator and customization settings, and read and update incidents. You need to know the account username and password.
- You have access to a Resilient integration server where you will deploy and run the functions code. If not, you need to install and configure the server as described in the [Integration Server Guide](#).

## Slack configuration

Prior to installing the Slack function, follow the Slack documentation (<https://api.slack.com/slack-apps>) to build a new Slack App.

In the *Basic Information* page, find the *Display Information* section as shown below, then configure your App name and upload the App icon. You may use the Resilient logo for your Slack App icon. `ResilientLogo.png` is included in the package and can be found in the `fn_slack/doc` directory. Make sure to save changes.

Your App's name is used for message authorship. How message authorship is attributed depends on a few factors, with some behaviors varying based on the kinds of tokens being used to post a message.

The App's name and icon is used as the author of the posted messages. Users can change message authorship in the app.config file or example workflows.

When uploading files, the name of the authenticated user of the Slack App is used for authorship. This behavior can be changed by adding a [Bot User](#).

## Display Information

This information will be shown in the Slack App Directory and in the Slack App  
For more information, view our [App Detail Guidelines](#).

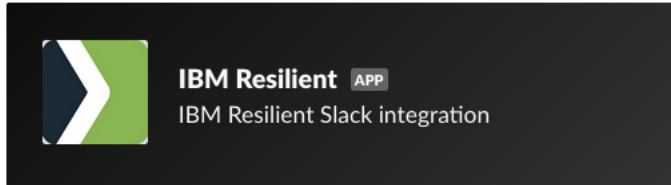
### App name

IBM Resilient

### Short description

IBM Resilient Slack integration

### App icon & Preview



### Background color

#000000

The Resilient integration requires that certain permissions are enabled. After you create a Slack App and set its Display Information, navigate to *OAuth & Permissions* page. Scroll down to the *Scopes* section and add the following permission scopes:

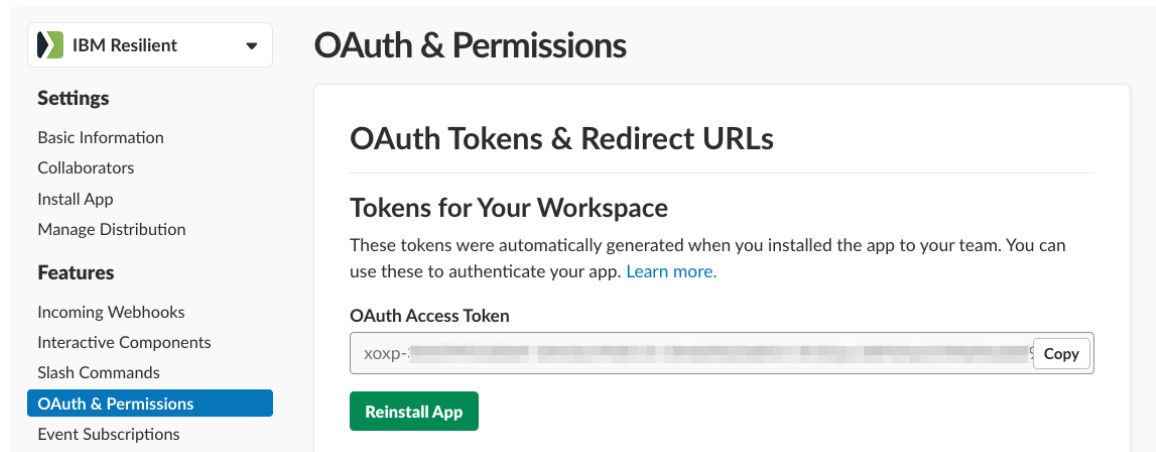
- channels:history -- to access user's public channels
- channels:read -- to access information about user's public channels
- channels:write -- to modify your public channels
- chat:write:bot -- to send messages as Slack App name
- chat:write:user -- to send messages as user
- groups:history -- to access content in user's private channels
- groups:read -- to access information about user's private channels

**NOTE:** The function only retrieves those private channels from your Slack workspace in which the Slack App's authorized user has been invited.

- groups:write -- to modify your private channels
- files:write:user -- to upload and modify files as user
- users:read -- to access your workspace's profile information
- users:read.email -- to view email addresses of people on this workspace

Make sure to save your changes and then either click on **Install App to Workspace** or **Reinstall App** button in the *OAuth Tokens & Redirect URLs* section and authorize changes in the next summary page.

Installation generates an **OAuth Access Token** as shown below.



## Install the Python components

The slack package contains Python components that will be called by the Resilient platform to execute the functions during your workflows. These components run in the 'resilient-circuits' integration framework.

The package also includes Resilient customizations that will be imported into the platform later.

Ensure that the environment is up to date,

```
sudo pip install --upgrade pip
sudo pip install --upgrade setuptools
sudo pip install --upgrade resilient-circuits
```

To install the package:

```
unzip app-fn_slack-x.x.x.zip
sudo pip install fn_slack-x.x.x.tar.gz
```

## Configure the Python components

The 'resilient-circuits' components run as an unprivileged user, typically named 'integration'. If you do not already have an 'integration' user configured on your appliance, create it now.

Perform the following to configure and run the integration:

1. Using sudo, become the integration user.

```
sudo su - integration
```

2. Use one of the following commands to create or update the resilient-circuits configuration file. Use `-c` for new environments or `-u` for existing environments.

```
resilient-circuits config -c
```

or

```
resilient-circuits config -u
```

3. Edit the resilient-circuits configuration file.

- a. In the [resilient] section, ensure that you provide all the information needed to connect to the Resilient platform.

- b. In the [fn\_slack] section, edit the settings as follows:

```
# Slack app OAuth Access Token
api_token=xoxp-xxxxxxxx-xxxxxxxxxxxx-xxxxxxxxxxxx-xxxxxxxxxxxx
```

```
# Username represents the default submission author.
# Used together with 'as_user=False'.
# You can also update the username on the Workflow.
username=Resilient
# template file override
#template_file=/var/rescircuits/slack_template.jinja2

# add proxy support here or use [integrations] for integration wide proxy
settings
#http_proxy=
#https_proxy=
```

## Deploy customizations to the Resilient platform

The package contains function definitions that you can use in workflows, and includes example workflows and rules that show how to use these functions.

1. Use the following command to deploy these customizations to the Resilient platform:

```
resilient-circuits customize
```

2. Respond to the prompts to deploy functions, message destinations, workflows and rules.

## Run the integration framework

To test the integration package before running it in a production environment, you must run the integration manually with the following command:

```
resilient-circuits run
```

The resilient-circuits command starts, loads its components, and continues to run until interrupted. If it stops immediately with an error message, check your configuration values and retry.

## Configure Resilient Circuits for restart

For normal operation, Resilient Circuits must run continuously. The recommend way to do this is to configure it to automatically run at startup. On a Red Hat appliance, this is done using a systemd unit file such as the one below. You may need to change the paths to your working directory and app.config.

1. The unit file must be named `resilient_circuits.service` To create the file, enter the following command:

```
sudo vi /etc/systemd/system/resilient_circuits.service
```

2. Add the following contents to the file and change as necessary:

```
[Unit]
Description=Resilient-Circuits Service
After=resilient.service
Requires=resilient.service

[Service]
Type=simple
User=integration
WorkingDirectory=/home/integration
ExecStart=/usr/local/bin/resilient-circuits run
Restart=always
TimeoutSec=10
Environment=APP_CONFIG_FILE=/home/integration/.resilient/app.config
```

```
Environment=APP_LOCK_FILE=/home/integration/.resilient/resilient_circuits.lock
[Install]
WantedBy=multi-user.target
```

3. Ensure that the service unit file is correctly permissioned, as follows:

```
sudo chmod 664 /etc/systemd/system/resilient_circuits.service
```

4. Use the systemctl command to manually start, stop, restart and return status on the service:

```
sudo systemctl resilient_circuits [start|stop|restart|status]
```

You can view log files for systemd and the resilient-circuits service using the journalctl command, as follows:

```
sudo journalctl -u resilient_circuits --since "2 hours ago"
```

## Test the Integration

Many integrations come with a self-test capability. This feature is enabled in the integration and with resilient-circuits version 30.0.111 or greater. Once the integration is configured in the resilient-circuits configuration file, testing to the end-point solution can be performed with the following command:

```
resilient-circuits selftest [-l fn-slack]
```

The resulting file will produce a result indicating success, failure or a message indicating that the feature is not available. Here are a few examples:

```
resilient-circuits selftest -l fn-slack
selftest: success, Elapsed time: 0.000000 seconds
```

## Function Descriptions

Once the function package deploys, you can view the three functions in the Resilient platform Functions tab, as shown below.

Customization Settings

Layouts Rules Scripts Workflows **Functions** Message Destinations Phases & Tasks Incident Types Breach Artifacts

Functions New Function

Search...

Name	Description	
<a href="#">Archive Slack Channel</a>	Function exports conversation history from Slack channel to a text file, saves the text file as an attachment and archives the Slack channel.	
<a href="#">Post message to Slack</a>	Function sends a message from an Incident, Task, Note or an Artifact to a Slack channel.	
<a href="#">Post attachment to Slack</a>	Function uploads Incident, Task or Artifact attachment to Slack channel.	

© Copyright IBM Corporation 2018

The package includes example workflows and rules that show how the functions can be used. You can copy and modify these workflows and rules for your own needs.

Customization Settings

Layouts Rules Scripts **Workflows** Functions Message Destinations Phases & Tasks Incident Types Breach Artifacts

Workflows New Workflow

Search...

Workflow Name	Description	Object Type	Rules	
<a href="#">Example: Archive Incident Slack Channel</a>	Exports conversation history from Incident associated Slack channel to a text file, saves the text file as an Attachment and archives the Slack channel.	Incident	<a href="#">Example: Archive Incident Slack Channel</a>	
<a href="#">Example: Archive Task Slack Channel</a>	Exports conversation history from Task associated Slack channel to a text file, saves the text file as an Attachment and archives the Slack channel.	Task	<a href="#">Example: Archive Task Slack Channel</a>	
<a href="#">Example: Post Artifact Attachment to Slack</a>	Upload Artifact Attachment to your Slack channel with an optional custom text message.	Artifact	<a href="#">Example: Post Artifact Attachment to Slack</a>	
<a href="#">Example: Post Artifact to Slack</a>	Post a message from the Artifact to your Slack channel. Send specifics about the Artifact with an optional custom text message.	Artifact	<a href="#">Example: Post Artifact to Slack</a>	
<a href="#">Example: Post Incident / Task Attachment to Slack</a>	Upload Incident or Task Attachment to your Slack channel with an optional custom text message.	Attachment	<a href="#">Example: Post Incident / Task Attachment to Slack</a>	
<a href="#">Example: Post Incident to Slack</a>	Post a message from the Incident to your Slack channel. Send specifics about the Incident with an optional custom text message.	Incident	<a href="#">Example: Post Incident to Slack</a>	
<a href="#">Example: Post Note to Slack</a>	Post a message from the Note to your Slack channel. Send specifics about the Incident or Task Note with an optional custom text message.	Note	<a href="#">Example: Post Note to Slack</a>	
<a href="#">Example: Post Task to Slack</a>	Post message from a Task to your Slack channel. Send specifics about the Task with an optional custom text message.	Task	<a href="#">Example: Post Task to Slack</a>	

© Copyright IBM Corporation 2018

## Function: Post message to Slack

This function posts data from an Incident, Task, Note, or an Artifact to a Slack channel, and takes the following input fields:

- **slack\_channel**: Name of the existing or a new Slack channel, where the app will post data. Channel names can contain only lowercase letters, numbers, hyphens, and underscores, and must be 21 characters or less. If you leave this field empty, the function tries to use the **slack\_channel** associated with the Incident or Task found in the Slack Conversations Data Table. If one is not defined, the workflow terminates.
- **slack\_is\_channel\_private**: Determines if the channel you are posting to should be private.
- **slack\_participant\_emails**: Comma-separated list of emails belonging to Slack users in your workspace that will be added to your channel.
- **slack\_text**: A text message or a container field to retain JSON fields to send to Slack.
- **slack\_mrkdwn**: Disables Slack markup parsing by setting to False.
- **slack\_as\_user**: If True, the authenticated user of the Slack App appears as the author of the message, ignoring any values provided in **slack\_username**.
- **slack\_username**: Replaces your Slack App's name to appear as the author of the message. Must be used in conjunction with **slack\_as\_user** set to False; otherwise, it is ignored.

### Customization Settings

**Functions / slack\_post\_message**

**Name \*** Post message to Slack

**API Name \*** slack\_post\_message

**Message Destination \*** slack

**Description** Function sends a message from an Incident, Task, Note or an Artifact to a Slack channel.

**Creator** [User Avatar]

**Last Modified** 11/28/2018 15:24

**Last Modified By** [User Avatar]

**Associated Workflows**

- Example: Post Artifact to Slack
- Example: Post Incident to Slack
- Example: Post Note to Slack
- Example: Post Task to Slack

**Inputs**

- slack\_channel
- slack\_is\_channel\_private
- slack\_participant\_emails
- slack\_text
- slack\_mrkdwn
- slack\_as\_user
- slack\_username
- incident\_id
- task\_id

**Input Fields**

Search...

- artifact\_id
- attachment\_id
- incident\_id
- slack\_as\_user
- slack\_channel
- slack\_is\_channel\_private
- slack\_mrkdwn
- slack\_participant\_emails
- slack\_text



The following example workflows call this function to post object data to a Slack channel:

- Example: Post Incident to slack
- Example: Post Task to Slack
- Example: Post Note to Slack
- Example: Post Artifact to Slack

The workflows can set the input field values from the Input tab.

The screenshot shows the 'Customization Settings' window for a workflow. The 'Workflows' tab is selected, showing the workflow 'Example: Post Incident to Slack'. The workflow details include:

- Name:** Example: Post Incident to Slack
- API Name:** create\_slack\_message
- Description:** Post a message from the Incident to your Slack channel. Send specifics about the Incident with an optional custom text
- Object Type:** Incident
- Creator:** [User Icon]
- Last Modified:** 12/03/2018 08:42
- Last Modified By:** [User Icon]
- Associated Rules:** Example: Post Incident to Slack

Below the details is a visual workflow diagram showing a sequence of steps: 'Input' -> 'Pre-Process Script' -> 'Output' -> 'Post-Process Script'. A tooltip indicates: 'Select the slack channel to post in and adjust the posting parameters as needed.' The 'Post-Process Script' step is highlighted with a yellow box.

The bottom section of the window is the 'Input' tab, which contains a table of input parameters and their values:

Input Parameter	Value
slack_channel	
slack_is_channel_private	Unknown
slack_participant_emails	
slack_text	
slack_mrkdwn	Unknown
slack_as_user	Unknown
slack_username	
incident_id	
task_id	

The default settings for posting messages in the function are:

- `parse="none"`: Slack does not perform any processing on the message. It keeps all markup formatting.
- `link_names=1`: Slack converts URLs, channel names (starting with a '#') and username IDs (starting with '<@ user\_id >') to links. The function can send messages like, "Hey user <@UCNC5K34J> check out #random", to Slack and it finds and links channel names and usernames.

Refer to the [Slack API chat.postMessage documentation](#) for a detailed explanation of the Slack arguments.

In the workflow's Pre-Process Script tab, the data to post is customizable. A flexible JSON structure is used to define the Incident, Task, Note and Artifact fields to post in Slack. An additional text message, set in the Menu Item, can also be included with the data posted. Fields can be removed and added following the defined data structure.

### Customization Settings

Layouts
Rules
Scripts
**Workflows**
Functions
Message Destinations
Phases & Tasks
Incident Types
Breach
Artifacts

Workflows / Example: Post Incident to Slack

Cancel
Save & Close
Save

Name \*
Example: Post Incident to Slack

API Name \* ⓘ
create\_slack\_message

Description
Post a message from the Incident to your Slack channel. Send specifics about the Incident with an optional custom text

Object Type \*
Incident

Creator
Last Modified
12/03/2018 08:42
Last Modified By
Associated Rules
Example: Post Incident

Input
Pre-Process Script
Output
Post-Process Script

Language: Python Theme light Mode Default Tab Size 2 - Font + Font

```

16 # Slack text message in JSON format
17 # -----
18 # Do not remove first 3 elements "Additional Text", "Resilient URL" and "Type of data",
19 # the information is used to generate the title of the message.
20 #
21 # Add/remove information using the syntax:
22 # "label": {{"type": "[string|richtext|boolean|datetime", "data": "resilient field data" }}
23 #
24 # Make sure to send "datetime" types as integers and not strings:
25 # without double quotes: { "type": "datetime", "data": resilient datetime data}
26 #
27 # Text fields like 'incident name', 'description' or 'Slack additional text message' can include double quotes.
28 # Watch out for embedded double quotes in these text fields and escape with field.replace(u'', u'\\'') otherwise json.loads will fail.
29 # slack_text = u''''''
30 "Additional Text": {{"type": "string", "data": "{0}" }},
31 "Resilient URL": {{"type": "incident", "data": "{1}" }},
32 "Type of data": {{"type": "string", "data": "{2}" }},
33 "Incident ID": {{"type": "string", "data": "{3}" }},
34 "Incident name": {{"type": "string", "data": "{4}" }},
35 "Description": {{"type": "richtext", "data": "{5}" }},
36 "Incident Types": {{"type": "string", "data": "{6}" }},
37 "NIST Attack Vectors": {{"type": "string", "data": "{7}" }},
38 "Confirmed": {{"type": "boolean", "data": "{8}" }},
39 "Date Created": {{"type": "datetime", "data": "{9}" }},
40 "Date Occurred": {{"type": "datetime", "data": "{10}" }},
41 "Date Discovered": {{"type": "datetime", "data": "{11}" }},
42 "Severity": {{"type": "string", "data": "{12}" }}
43 }'''.format(
44     rule_additional_text.replace(u'', u'\\''),
45     incident_id_str,
46     "Incident",
47     incident_id_str,

```

Some of the function input fields can be set by clicking a Menu Item rule for the object type.

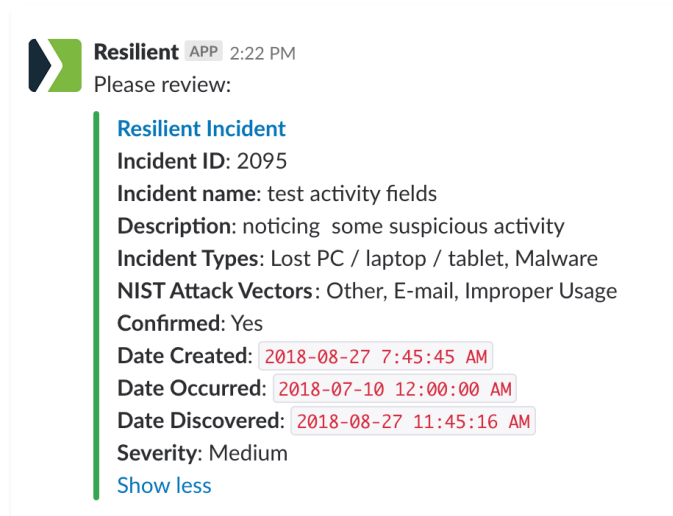
The screenshot shows the Freitag incident management interface. At the top, the header 'Freitag' is visible. On the right, an 'Actions' dropdown menu is open, displaying a list of actions: 'Example: Archive Incident Slack Channel', 'Example: Post Incident to Slack' (highlighted with a red circle), 'Action Status', 'Workflow Status', 'Close Incident', and 'Delete Incident'. The main interface is divided into several sections: 'Summary' (ID 2111, Phase Respond, Severity Low, Date Created 11/30/2018, Date Occurred —, Date Discovered 11/30/2018, Was personal information or personal data involved? Unknown, Incident Type —), 'Description' (No description.), 'Tasks' (0% Complete, Filter: Active, Selected, Add Task), 'Respond' (Respond - (Data Breach - Organizational)), and 'People' (Created By). The 'Tasks' section includes a table with columns: Task Name, Owner, Due Date, and Flags. The 'Respond' section includes a table with columns: Respond - (Data Breach - Organizational), Unassigned, No due date, and a chat icon.

These values either override those set in the workflow or are included with the message posted in Slack:

The screenshot shows a dialog box titled 'Example: Post Incident data to Slack'. It contains four input fields: 'Slack channel name' (Existing or new Slack channel name), 'Slack is channel private' (Unknown), 'Slack users emails' (slack.user@email.com, slack.user2@), and 'Additional text' (Please review posted Resilient Data). At the bottom, there are 'Cancel' and 'Execute' buttons.

- The values in Slack channel name, Slack is channel private and Slack user emails, override those set in the workflow.
- Additional text: Includes the text in this field with the Incident, Note, Artifact, Attachment or Task data to send to Slack.

Your posted message will look like this in Slack:



## Function: Post attachment to Slack

This function uploads attachments from an Incident, Task, or Artifact to a Slack channel.

Customization Settings

Layouts Rules Scripts Workflows **Functions** Message Destinations Phases & Tasks Incident Types Breach Artifacts

Functions / slack\_post\_attachment

**Name \*** Post attachment to Slack

**API Name \*** slack\_post\_attachment

**Message Destination \*** slack

**Description** Function uploads Incident, Task or Artifact Attachments to Slack channel.

**Creator** [User Icon]

**Last Modified** 11/28/2018 15:24

**Last Modified By** [User Icon]

**Associated Workflows** Example: Post Artifact Attachment to Slack  
Example: Post Incident / Task Attachment to Slack

**Inputs**

- slack\_channel
- slack\_is\_channel\_private
- slack\_participant\_emails
- slack\_text
- incident\_id
- task\_id
- artifact\_id
- attachment\_id

**Input Fields**

Search...

- artifact\_id
- attachment\_id
- incident\_id
- slack\_as\_user
- slack\_channel
- slack\_is\_channel\_private
- slack\_mrkdwn
- slack\_participant\_emails
- slack\_text

The workflows associated with this function are:

- Example: Post Incident/Task Attachment to Slack
- Example: Post Artifact Attachment to Slack

The function input fields can be set on the Input Tab of the workflow or when you click the Menu Item for the object, which is the same way as shown for the Post message to Slack Function.

### Description

No description.

Tasks	Details	Breach	Notes	Members	News Feed	<b>Attachments</b>	Stats	Timeline	Artifacts
tab [redacted]									

### Attachments

Drag file here

Upload File

Maximum file size: 25 MB

Search...

☒ Show Task Attachments

Uploaded By: All

Date Created: All ▾

Type	Name	Uploaded By	Date Added	Size	Actions
	mytest_incident_attachment.txt	Fobs Lobs	12/01/2018	556 bytes	
	[redacted]				

Example: Post Incident / Task Attachment to Slack

Disolavina 1 - 2 of 2

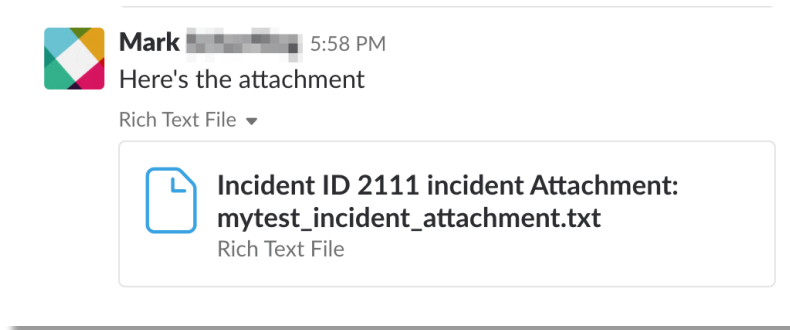
### Example: Post Incident / Task Attachment to Slack

Slack channel name ⓘ	<input type="text" value="my_test_slack_channel"/>
Slack is channel private ⓘ	<input type="text" value="Unknown"/>
Slack users emails ⓘ	<input type="text" value="resilientuser@ibm.com"/>
Additional text ⓘ	<input type="text" value="Here's the attachment"/>

Cancel

Execute

Your posted message with an upload attachment will look like this in Slack:



To upload files, the function uses the [Slack API files.upload method](#), which uses the Slack App OAuth Token that belongs to the authenticated user of the Slack App. File uploads sent using this token are uploaded on behalf of the user - not the Slack App. This behavior can be changed by adding a [Bot User](#) to your Slack App and using Bot User OAuth Token for file uploads.

## Function: Archive Slack Channel

This function exports a conversation history from a Slack channel to a text file, saves the text file as a Resilient attachment, and archives the Slack channel.

### Customization Settings


A screenshot of the 'Customization Settings' for the 'slack\_archive\_channel' function. The interface has a top navigation bar with tabs: Layouts, Rules, Scripts, Workflows, Functions (selected), Message Destinations, Phases & Tasks, Incident Types, Breach, and Artifacts. Below the tabs, there's a breadcrumb 'Functions / slack\_archive\_channel' and buttons for 'Cancel', 'Save & Close', and 'Save'. The main settings area includes: 'Name' (Archive Slack Channel), 'API Name' (slack\_archive\_channel), 'Message Destination' (slack), and 'Description' (Function exports conversation history from Slack channel to a text file, saves the text file as an Attachment and archives the Slack channel.). On the right, there's a 'Creator' field, 'Last Modified' (11/28/2018 15:24), 'Last Modified By', and 'Associated Workflows' (Example: Archive Incident Slack Channel, Example: Archive Task Slack Channel). At the bottom, there's an 'Inputs' section with 'incident\_id' and 'task\_id' fields, and an 'Input Fields' section with a search bar and a list of fields: artifact\_id, attachment\_id, incident\_id, slack\_as\_user, slack\_channel, slack\_is\_channel\_private, slack\_mrkdwn, slack\_participant\_emails, and slack\_text.

The following example workflows demonstrate archiving a Slack channel from an Incident or Task:

- Example: Archive Incident Slack Channel
- Example: Archive Task Slack Channel

When invoked, the Archive Slack Channel function searches for the associated channel name in the Slack Conversations Data Table. If an Incident or Task has an existing connection with a slack\_channel, the function archives this channel.

The exported conversation history looks like this:



```
slackmsgexportchanneldemo-2103-incident.txt
1 - Resilient POSTED ON 2018-11-01 14:41:54:
This channel has been set to be archived from Resilient.

2 - [REDACTED] POSTED ON 2018-11-01 14:41:20:
I'll take a look asap.

3 - Resilient POSTED ON 2018-11-01 14:35:13:
*Incident ID*: 2103
*Artifact Type*: IP Address
*Artifact Value*: 192.168.56.3
*Artifact Created By*: Resilient Sysadmin
*Artifact Created on*: `<!date^1541066628^{date_num} {time_secs}|2018-11-01 10:03:48>`

2 REPLIES:
4 - [REDACTED] POSTED A REPLY ON 2018-11-01 14:41:25:
I'll take a look asap.

5 - [REDACTED] POSTED A REPLY ON 2018-11-01 14:41:34:
File name [REDACTED]
File url [REDACTED]
```

## Resilient Platform Configuration

The following eight rules, which you can customize, are defined in the package:

### Rules

[New Rule -](#)

Order	Rule Name	Process Type	Object Type	Conditions	
-	Example: Archive Incident Slack Channel	Menu Item	Incident		
-	Example: Archive Task Slack Channel	Menu Item	Task		
-	Example: Post Artifact Attachment to Slack	Menu Item	Artifact	Type	
-	Example: Post Artifact to Slack	Menu Item	Artifact	Type	
-	Example: Post Incident / Task Attachment to Slack	Menu Item	Attachment		
-	Example: Post Incident to Slack	Menu Item	Incident		
-	Example: Post Note to Slack	Menu Item	Note		
-	Example: Post Task to Slack	Menu Item	Task		

NOTE: The rule, Example: Post Artifact Attachment to Slack, allows uploading attachments from only certain types of artifacts. Artifact type `Malware sample` is excluded because of the potential risk of sending malware samples to your Slack channel.

### Customization Settings

Layouts

**Rules**

Scripts

Workflows

Functions

Message Destinations

Phases & Tasks

Incident Types

Breach

Artifacts

Rules / Example: Post Artifact Attachment to Slack

Cancel

Save & Close

Save

Display Name \*

Example: Post Artifact Attachment to Slack

Object Type

Artifact

Conditions

Add conditions in which to invoke the rule. [Clear All](#)

Type

has one of

Email Attachment X

Log File X

Other File X

RFC 822 Email Message File X

X509 Certificate File X



There is also a Slack Conversations Data Table, `slack_conversations_db`, created in the `resilient-circuits` `customize` step. Users may add this data table to a custom layout.

Slack

Edit

Slack Conversations

Search...

Print

Export

Time	Resilient ID	Slack channel name	Slack channel type	Slack URL	
12/05/2018 13:27:58	RES-2095	incident-2095	Private	<a href="#">Link</a>	...
12/05/2018 13:30:05	RES-2095-2251221	task-sandbox-2095	Public	<a href="#">Link</a>	...
12/05/2018 13:31:14	RES-2095-2251203	task-triage-2095	Private	<a href="#">Link</a>	...

Displaying 1 - 3 of 3

The purpose of the data table is to save connections between an Incident and a Slack channel or a Task and a Slack channel. An Incident and each Task can each have one associated channel – the default channel, but users may also post to other channels – separate non-default channels. The function only saves the connection to the first Slack channel that gets a post.

When invoking the Post message to Slack or Post attachment to Slack functions, users can specify whether they are posting to their associated - default Slack channel or to a separate - non-default channel.

If the function input field, `channel_name`, is not specified either on the Menu Item or on the Input tab of the workflow, the function searches for the associated channel name in the Slack Conversations Data Table. If the Incident or Task has an existing connection with a `slack_channel`, the function finds this channel in your workspace and posts there.

If `channel_name` input field is specified, the function tries to find this channel in your workspace and posts there. If the channel does not yet exist, it creates a new channel.

If `channel_name` input field is specified and the Incident or Task already has an existing connection with a `slack_channel`, the function ignores the associated one in the data table and posts to the input one.

## Troubleshooting

There are several ways to verify the successful operation of a function.

- Resilient Action Status

When viewing an incident, use the Actions menu to view Action Status. By default, pending actions and errors are displayed. Modify the filter for actions to also show Completed actions. Clicking on an action displays additional information on the progress made or what error occurred.

- Resilient Scripting Log

A separate log file is available to review scripting errors. This is useful when issues occur in the pre-processing or post-processing scripts. The default location for this log file is:

`/var/log/resilient-scripting/resilient-scripting.log`.

- Resilient Logs

By default, Resilient logs are retained at `/usr/share/co3/logs`. The `client.log` may contain additional information regarding the execution of functions.

- Resilient-Circuits

The log is controlled in the `.resilient/app.config` file under the section `[resilient]` and the property `logdir`. The default file name is `app.log`. Each function will create progress information. Failures will show up as errors and may contain python trace statements.

## Support

For additional support, go to <https://www.ibm.com/mysupport>.

Including relevant information from the log files will help us resolve your issue.

## Template file

If you'd like to customize the template file, use the `template_file` app.config setting to specify the location of your jinja2 template. Below is the default template.

```
{%- if is_msg_parent -%}
  {{- number }} - {{ username }} POSTED ON {{ msg_time }}:{{ '\n' -}}
  {%- if msg_pretext -%}
    {{- msg_pretext }}{{ '\n' -}}
  {%- endif -%}
  {%- if msg_text -%}
    {{- msg_text }}{{ '\n' -}}
  {%- endif -%}
  {%- if file_name -%}
    {{- file_name }}{{ '\n' -}}
  {%- endif -%}
  {%- if file_permalink -%}
    {{- file_permalink }}{{ '\n' -}}
  {%- endif -%}
  {%- if reply_count and reply_count > 0 -%}
    {{- reply_count }} REPLIES:{{ '\n' -}}
  {%- endif -%}
{%- else -%}
  {{- '\t' }}{{ number }} - {{ username }} POSTED A REPLY ON {{ msg_time
```

```
}}:{{ '\n' -}}  
  {%- if msg_text -%}  
    {{- '\t' }}{{ msg_text }}{{ '\n' -}}  
  {%- endif -%}  
  {%- if file_name -%}  
    {{- '\t' }}{{ file_name }}{{ '\n' -}}  
  {%- endif -%}  
  {%- if file_permalink -%}  
    {{- '\t' }}{{ file_permalink }}{{ '\n' -}}  
  {% endif %}{{ '\n' -}}  
{%- endif -%}
```