

# IBM Resilient



## Security Orchestration, Automation and Response Platform

Carbon Black Protection Integration V1.0.2

Release Date: August 2019

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 then returns the results to the workflow. The results can be acted upon by scripts, rules, and workflow decision points to dynamically orchestrate the security incident response activities.

This guide describes the Carbon Black Protection Integration.

## Overview

This integration consists of 12 functions which call various APIs to perform different actions, such as retrieving approval request details, updating approval requests and deleting files. It also contains a polling component to create incidents in the Resilient platform that correspond to approval requests in Carbon Black Protection.

## Installation

You download the function package to a Resilient integration server, and from there you deploy the functions and components to a Resilient platform. These procedures are provided in the [Resilient Integration Server Guide \(PDF\)](#).

The functions included this package have the following requirements, which are above and beyond those listed in the *Resilient Integration Server Guide*.

- Resilient platform is version 30 or later.
- Carbon Black Protection v8.1 or later.

After installing the package, Resilient Circuits creates a new section, [fn\_cb\_protection], in the app.config file. You need to edit the following settings in that section.

```
[fn_cb_protection]
# Name or IP address of your CbProtect server
server=10.200.1.1

# Access token issued by the CbProtect administrator
token= XXXX-XXXX-XXXX-XXXX

# If your CbProtect server has a self-signed TLS certificate, you cannot verify
it:
# verify_cert=false

# Interval (seconds) for automatic escalation of approval requests, set 0 to
disable
# Suggest 300 as a starting point, which will check CbProtect every 5 minutes
escalation_interval=0

# Optional: query for which requests to escalate; default is to escalate all
open approval requests
# escalation_query=resolution:0

# Optional: path to a custom template file for the escalated incident
# template_tile=/usr/integration/bit9_escalation.jinja

# Optional: set this to only escalate a single request ID, e.g. when testing a
custom template
# test_single_request=999
```

## Function Descriptions

Once the function package deploys the functions, you can view them in the Resilient platform Functions tab, as shown below. The package also includes example workflows and rules that show how the functions can be used. You can copy these workflows and rules for your own needs.

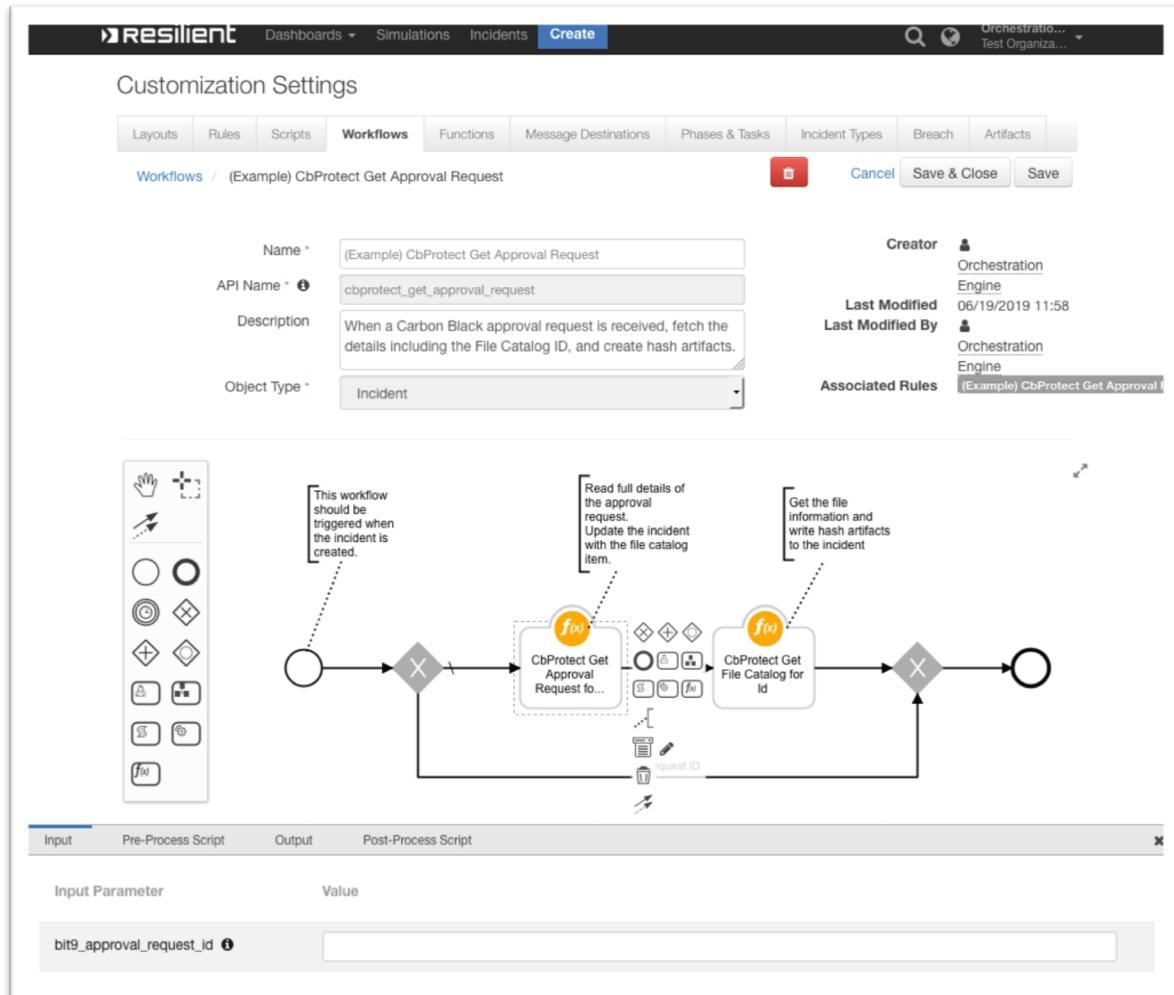
The screenshot shows the Resilient platform interface with the 'Functions' tab selected. The page title is 'Customization Settings' and the sub-section is 'Functions'. A search bar is at the top left, and a 'New Function' button is at the top right. Below the header is a horizontal navigation bar with tabs: Layouts, Rules, Scripts, Workflows, Functions (selected), Message Destinations, Phases & Tasks, Incident Types, Breach, and Artifacts. The main content area displays a table of functions:

Name	Description	Action
CbProtect Delete File	Deletes a file from provided systems.	trash
CbProtect Delete File Rule for Id	Delete a file rule.	trash
CbProtect Get Approval Request for Id	Get an approval request by ID.	trash
CbProtect Get Approval Request for Query Condition	Return approval requests that match the given criteria.	trash
CbProtect Get File Catalog for Id	Get a file catalog item by ID.	trash
CbProtect Get File Catalog for Query Condition	Return file catalog objects that match the given criteria.	trash
CbProtect Get File Instance for Query Conditions	Return file instance objects that match the given criteria.	trash
CbProtect Get File Rule for Id	Get a file rule by ID.	trash
CbProtect Get File Rule for Query Condition	Return file rules that match the given criteria.	trash
CbProtect Update Approval Request	Update an approval request.	trash
CbProtect Update File Instance Local State	Update the approval state of a file instance.	trash
CbProtect Update File Rule	Create or update a File Rule.	trash

At the bottom center of the page, there is a copyright notice: © Copyright IBM Corporation 2019.

## bit9\_approval\_request\_get: CbProtect Get Approval Request for Id

Given an approval request's ID, the function returns the details of the approval request. The function takes one input, bit9\_approval-request\_id, which is a number. The following is an example of this function in the (Example) CbProtect Get Approval request workflow.



## bit9\_approval\_request\_query: CbProtect Get Approval Request for Query Condition

This function takes one input, bit9\_query which is a query string, and returns the approval requests that match the given query condition. The following is an example of this function in the (Example) CbProtect Get Appr Req for Q 'fileName:notepad.exe' workflow. You can set a different query condition following the guidelines

<https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#query-condition> and review the all approval request properties to query here

<https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#approvalrequest>.  
fileName:notepad.exe represents name of the file on the agent.

Customization Settings

Workflows / (Example) CbProtect Get Appr Req for Q 'fileName:notepad.exe'

Name: (Example) CbProtect Get Appr Req for Q 'fileName:notepad.exe'

API Name: example\_cbprotection\_query\_approval\_request

Description: Queries for approval requests based on the provided query.

Object Type: Incident

Creator: Orchestration Engine 06/19/2019 11:57 Last Modified By: Orchestration Engine

Associated Rules: (Example) CbProtect Get Appro

Start your workflow here

Queries for approval requests with the file name notepad.exe

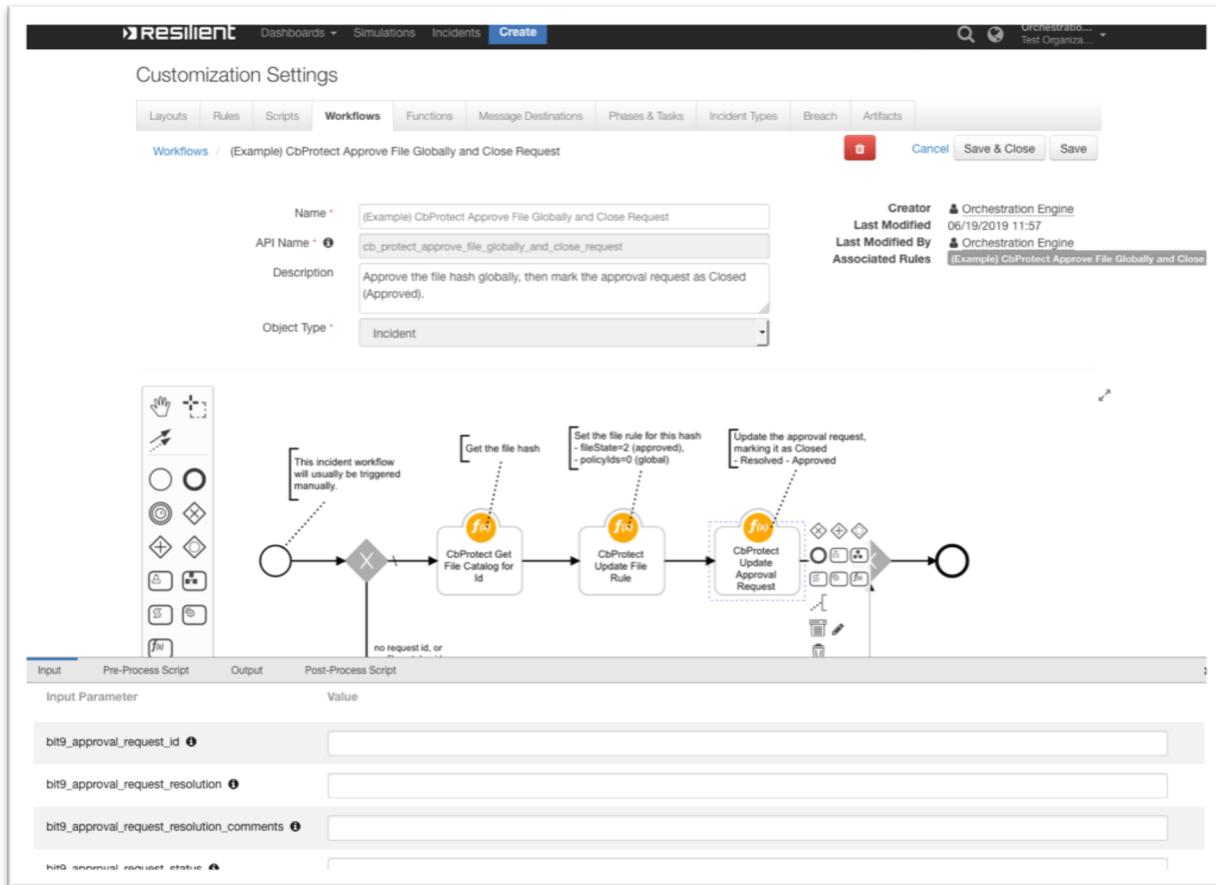
A note is created with the results

CbProtect Get Approval Request fo...

Input Parameter: bit9\_query Value: fileName:notepad.exe

## bit9\_approval\_request\_update: CbProtect Update Approval Request

This function accepts as input a request ID, approval request resolution, comments, and status. With these, it updates an approval request. The following is an example of this function in a workflow:



## bit9\_file\_delete: CbProtect Delete File

This function deletes a file from one or all computers using Carbon Black Protection. Set the bit9\_file\_action input to delete by file hash or file name (choose file hash to set catalog ID). Set the bit9\_computer\_id input for a specific computer, or use "0" to select all computers. Then, depending on the chosen file action, set the catalog ID, file hash, or file name. The following is an example of this function in the (Example) CbProtect Delete File By Hash workflow:

The screenshot shows the Resilient platform's 'Customization Settings' interface for a workflow. The workflow is titled '(Example) CbProtect Delete File By Hash'. The configuration includes:

- Name:** (Example) CbProtect Delete File By Hash
- API Name:** example\_cbprotect\_delete\_file
- Description:** Deletes a file by hash of type SHA1, SHA256 and MD5 from all systems.
- Object Type:** Artifact
- Creator:** Orchestration Engine
- Last Modified:** 06/20/2019 16:24
- Last Modified By:** Orchestration Engine
- Associated Rules:** (Example) CbProtect Delete File B

The workflow diagram consists of a start node, a central 'CbProtect Delete File' activity node, and an end node. Annotations provide instructions for setting up the workflow:

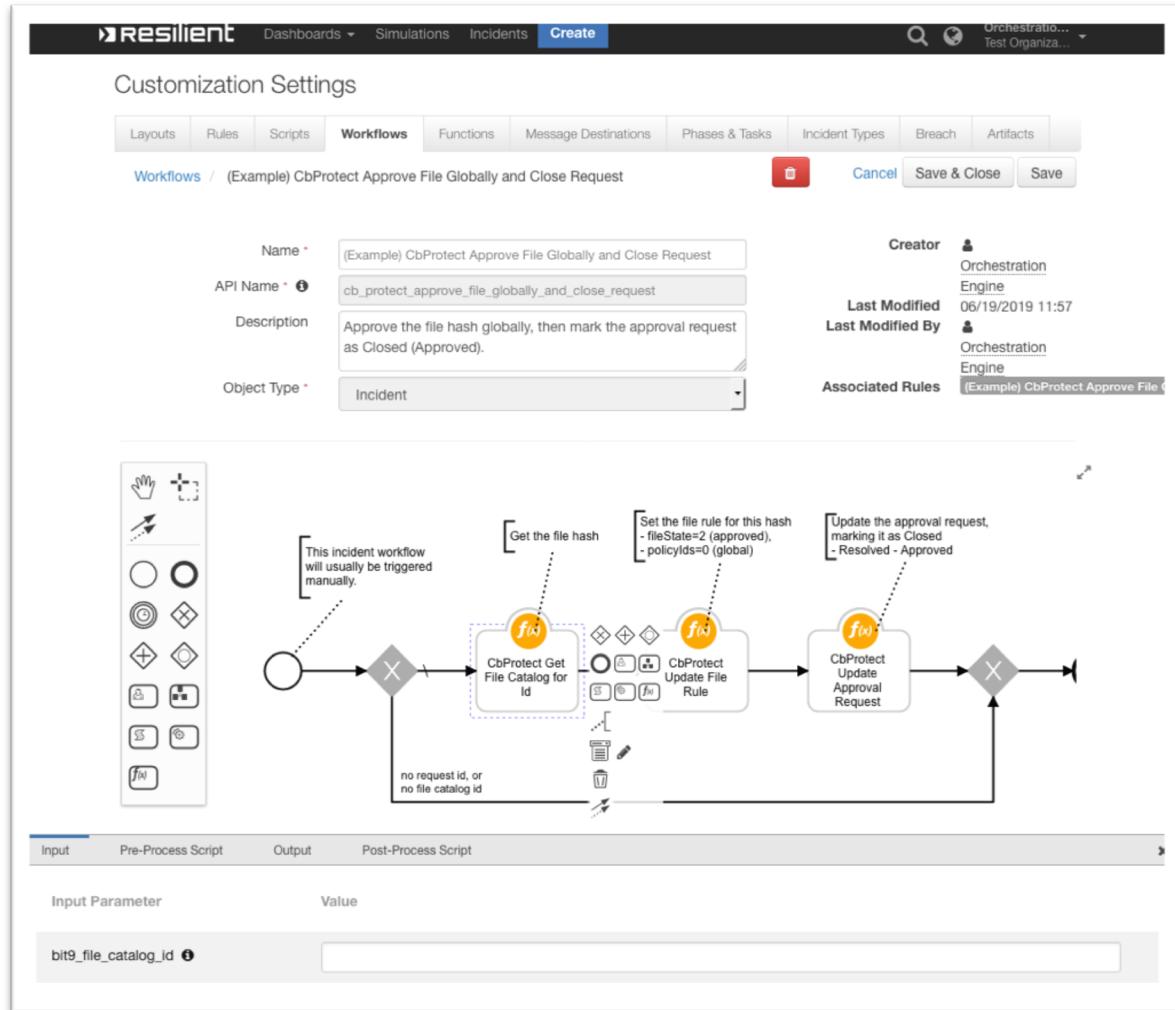
- A note 'Start your workflow here' points to the start node.
- A note 'Update inputs with the hash value for the file you wish to delete' points to the 'bit9\_file\_action' input field.
- A note 'A note is created with the results' points to the 'Post-Process Script' section.

The 'Input' section lists the parameters:

Input Parameter	Value
bit9_file_action ⓘ	DeleteFileByHash
bit9_computer_id ⓘ	
bit9_file_catalog_id ⓘ	
bit9_file_hash ⓘ	

## bit9\_file\_catalog\_get: CbProtect Get File Catalog for Id

Returns the file catalog details based on the catalog ID provided. The following is an example of this function in the (Example) CbProtect Approve File Globally and Close Request workflow:



## bit9\_file\_catalog\_query: Cbprotect Get File Catalog for Query Condition

Returns file catalogs and their details from a provided query string. The following is an example of this function in (Example) CbProtect Get File Catalog for Query Condition workflow. You can set a different query condition following the guidelines

<https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#query-condition>  
and review the all file catalog properties to query here

<https://developer.carbonblack.com/reference/enterprise-protection/8.0/rest-api/#filecatalog>.

The screenshot shows the Resilient platform's 'Workflows' section. A new workflow is being created with the following details:

- Name:** (Example) CbProtect Get File Catalog for Query Condition
- API Name:** example\_cbprotection\_query\_file\_catalog
- Description:** Get the details of the file catalog matching the single Cb Protect query condition. Results of the query are written to an incident note. An example of a single query condition: 'trust:1'
- Object Type:** Incident

**Workflow Diagram:**

```
graph LR; Start((Start)) --> Activity[CbProtect Get File Catalog for Qu...]; Activity --> End((End));
```

**Annotations:**

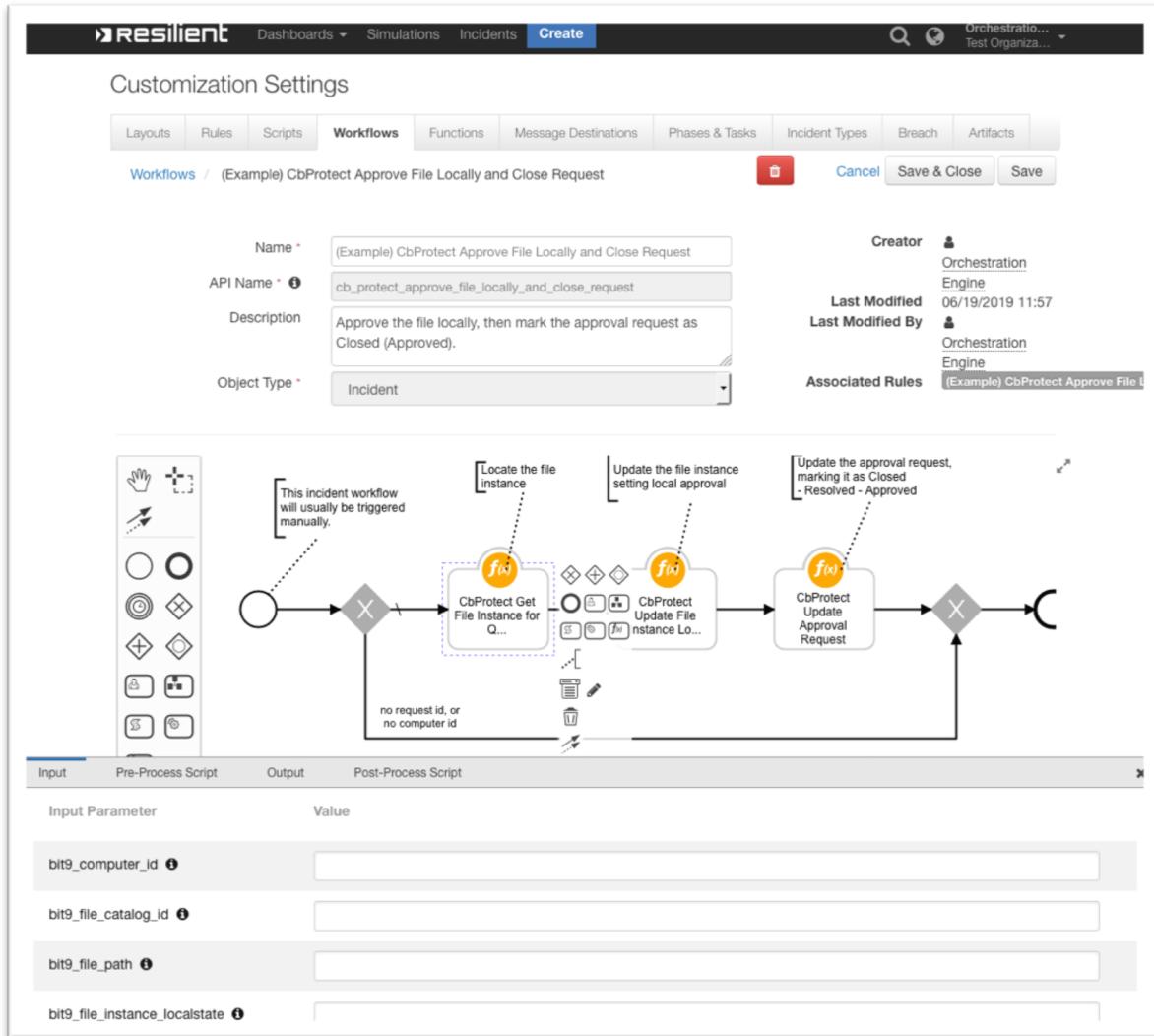
- Get the file catalog for files matching the result of the single Cb Protect query condition.
- An incident note is created containing the results.
- Start your workflow here.

**Post-Process Script (Python):**

```
1 content=u'Results from CbProtect Get File Catalog by Query Condition: {} items found\n{}'.format(results.count, results.pretty_results)
2
3 incident.addNote(helper.createPlainText(content))
4 # {'count': 656, 'items': [{"u'trust': 1, 'u'pathName': 'u'c:\\windows\\syswow64', 'u'fileName': 'u'ktmw32.dll', 'u'sha256': u'3c9fc4fb08d51c5024b3f5a2794b724ed17c3b45e434eb845478c3301
```

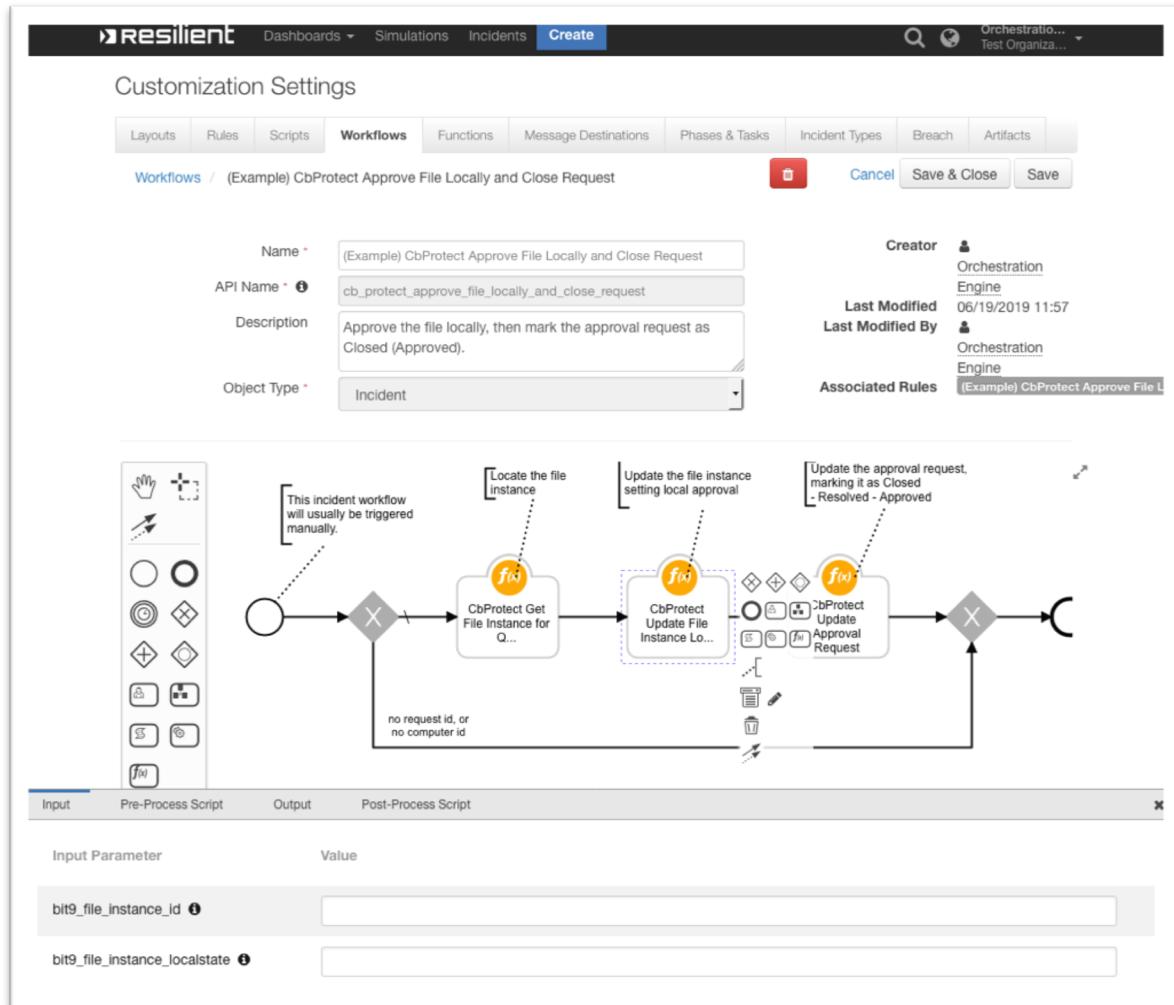
## bit9\_file\_instance\_query: CbProtect Get File Instance for Query Conditions

Returns file instance objects that match the given criteria from the inputs. The following is an example of this function in the (Example) CbProtect Approve File Locally and Close Request workflow:



## bit9\_file\_instance\_update: CbProtect Update File Instance Local State

Updates a file instance's local approval/banned setting. This function has inputs for the file instance ID and the local state (for example, approved = 2). The following includes an example of this function in the (Example) CbProtect Approve File Locally and Close Request workflow:



## bit9\_file\_rule\_update: CbProtect Update File Rule

This function updates a file rule in Carbon Black based on the data passed as inputs. The following is an example of this function in the (Example) CbProtect Approve File Globally and Close Request workflow:

Customization Settings

Workflows / (Example) CbProtect Approve File Globally and Close Request

Name: (Example) CbProtect Approve File Globally and Close Request  
API Name: cb\_protect\_approve\_file\_globally\_and\_close\_request  
Description: Approve the file hash globally, then mark the approval request as Closed (Approved).  
Object Type: Incident

Creator: Orchestration Engine  
Last Modified: 06/19/2019 11:57  
Last Modified By: Orchestration Engine  
Associated Rules: (Example) CbProtect Approve File

```
graph LR; Start(( )) --> IncidentX{ }; IncidentX -- "no request id, or no file catalog id" --> GetCatalogX{CbProtect Get File Catalog for Id}; GetCatalogX --> UpdateRuleX{CbProtect Update File Rule}; UpdateRuleX -- "Set the file rule for this hash - fileState=2 (approved), - policyIds=0 (global)" --> ApprovalX{CbProtect Update Approval Request}; ApprovalX -- "Update the approval request, marking it as Closed - Resolved - Approved" --> End(( ));
```

Input Parameter Value

bit9_file_rule_id	1
bit9_file_catalog_id	
bit9_file_rule_name	
bit9_file_rule_description	
bit9_file_rule_filestate	
bit9_file_rule_sourcetype	5
bit9_file_rule_policyids	
bit9_file_rule_hash	

## Carbon Black Protection Resilient Polling Component

This integration contains a polling component that automatically escalates approval requests into the Resilient platform. To enable this feature, the `escalation_interval` variable in the `app.config` file must be set to an integer greater than 0. This integer represents the interval in number of seconds for the automatic escalation of approval requests. It is recommended to start at 300, which checks every 5 mins.

You can also set optional values, such as `escalation_query`, which escalates approval requests that match the query; if not set, it defaults to all open approval requests. In addition, you can set `template_file` to the location of a custom jinja template file; if not set, the default template file is used. To create your own custom jinja file, you should use the default jinja file as a reference. This file can be found when expanding the package in the following directory:

```
fn_cb_protection-<version#>/fn_cb_protection/data/
```