fn_remedy

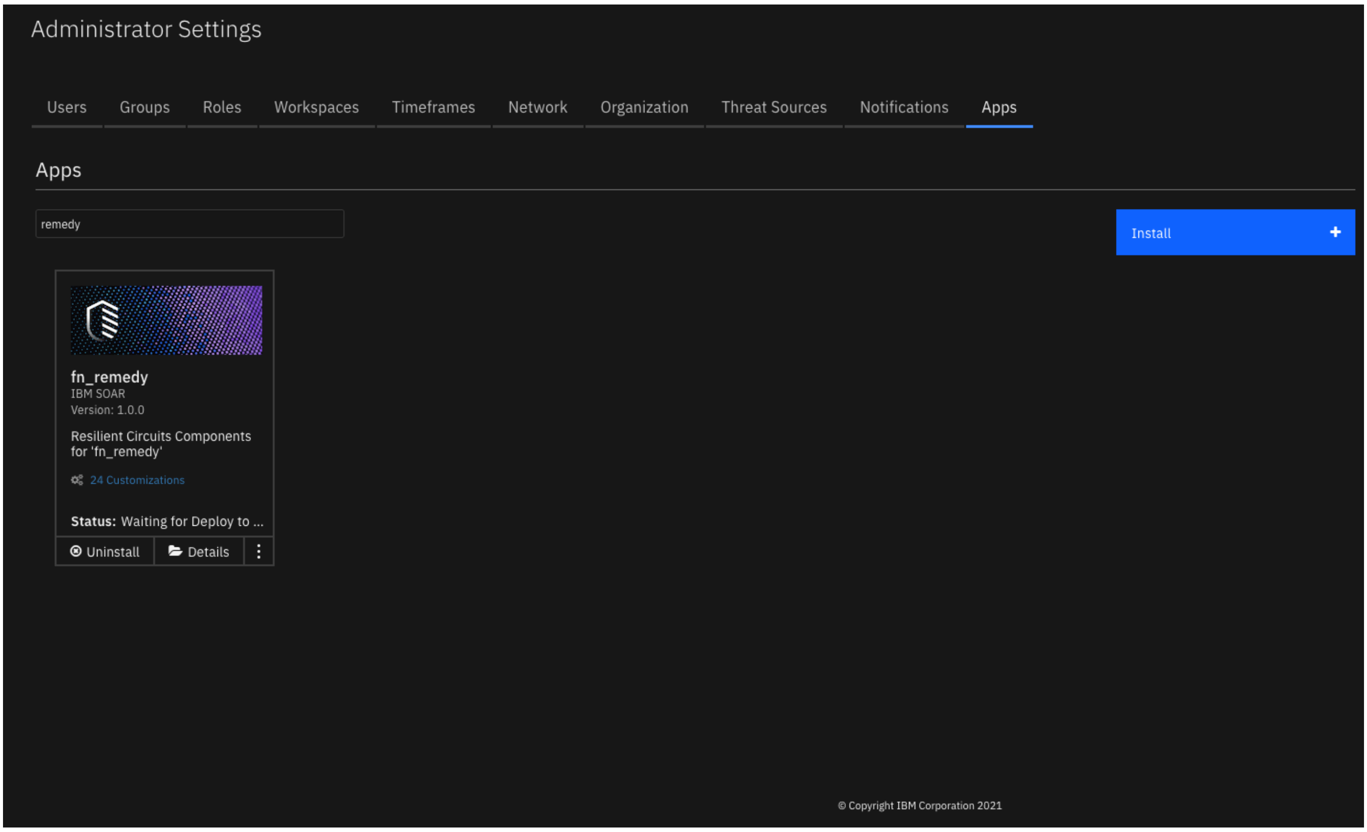
Table of Contents

Release Notes

Version	Date	Notes
1.0.0	04/2021	Initial Release

Overview

Resilient Circuits Components for 'fn_remedy'



remedy

fn_remedy
IBM SOAR
Version: 1.0.0

Resilient Circuits Components for 'fn_remedy'

24 Customizations

Status: Waiting for Deploy to ...

Resilient Circuits Components for 'fn_remedy.' This integration provides the capability to create new incidents in Remedy from Resilient tasks via the HPD:IncidentInterface_Create form over the REST API. Once the task is complete, this integration also provides the capability to close existing Remedy Incidents by updating their status to "Resolved."

Key Features

- Send CP4S Case tasks to Remedy as Incidents
- Close Remedy Incidents from CP4S

Requirements

This app supports the IBM Resilient SOAR Platform and the IBM Cloud Pak for Security.

Resilient platform

The Resilient platform supports two app deployment mechanisms, App Host and integration server.

If deploying to a Resilient platform with an App Host, the requirements are:

- Resilient platform >= `39.0.0`.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

If deploying to a Resilient platform with an integration server, the requirements are:

- Resilient platform >= `39.0.0`.
- The app is in the older integration format (available from the AppExchange as a `zip` file which contains a `tar.gz` file).
- Integration server is running `resilient-circuits>=30.0.0` and `resilient-lib>=39.0.0`.

- If using an API key account, make sure the account provides the following minimum permissions:

Name	Permissions
Org Data	Read
Function	Read
Incidents	Read
Incident Notes	Write

The following Resilient platform guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *Integration Server Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings.
- *System Administrator Guide*: provides the procedure to install, configure and deploy apps.

The above guides are available on the IBM Knowledge Center at ibm.biz/resilient-docs. On this web page, select your Resilient platform version. On the follow-on page, you can find the *App Host Deployment Guide* or *Integration Server Guide* by expanding **Resilient Apps** in the Table of Contents pane. The System Administrator Guide is available by expanding **System Administrator**.

Cloud Pak for Security

If you are deploying to IBM Cloud Pak for Security, the requirements are:

- IBM Cloud Pak for Security >= 1.4.
- Cloud Pak is configured with an App Host.
- The app is in a container-based format (available from the AppExchange as a `zip` file).

The following Cloud Pak guides provide additional information:

- *App Host Deployment Guide*: provides installation, configuration, and troubleshooting information, including proxy server settings. From the Table of Contents, select Case Management and Orchestration & Automation > **Orchestration and Automation Apps**.
- *System Administrator Guide*: provides information to install, configure, and deploy apps. From the IBM Cloud Pak for Security Knowledge Center table of contents, select Case Management and Orchestration & Automation > **System administrator**.

These guides are available on the IBM Knowledge Center at ibm.biz/cp4s-docs. From this web page, select your IBM Cloud Pak for Security version. From the version-specific Knowledge Center page, select Case Management and Orchestration & Automation.

Proxy Server

The app **does** support a proxy server.

Remedy Platform

This app requires Remedy IT Service Management Suite 20.x or above with AR Server 9.x or above. The REST API must be enabled and exposed on any port. If the REST API is not already enabled on the Remedy Platform, consult their documentation on Configuring the REST API.

Installation

Install

- To install or uninstall an App or Integration on the *Resilient platform*, see the documentation at ibm.biz/resilient-docs.
- To install or uninstall an App on *IBM Cloud Pak for Security*, see the documentation at ibm.biz/cp4s-docs and follow the instructions above to navigate to Orchestration and Automation.

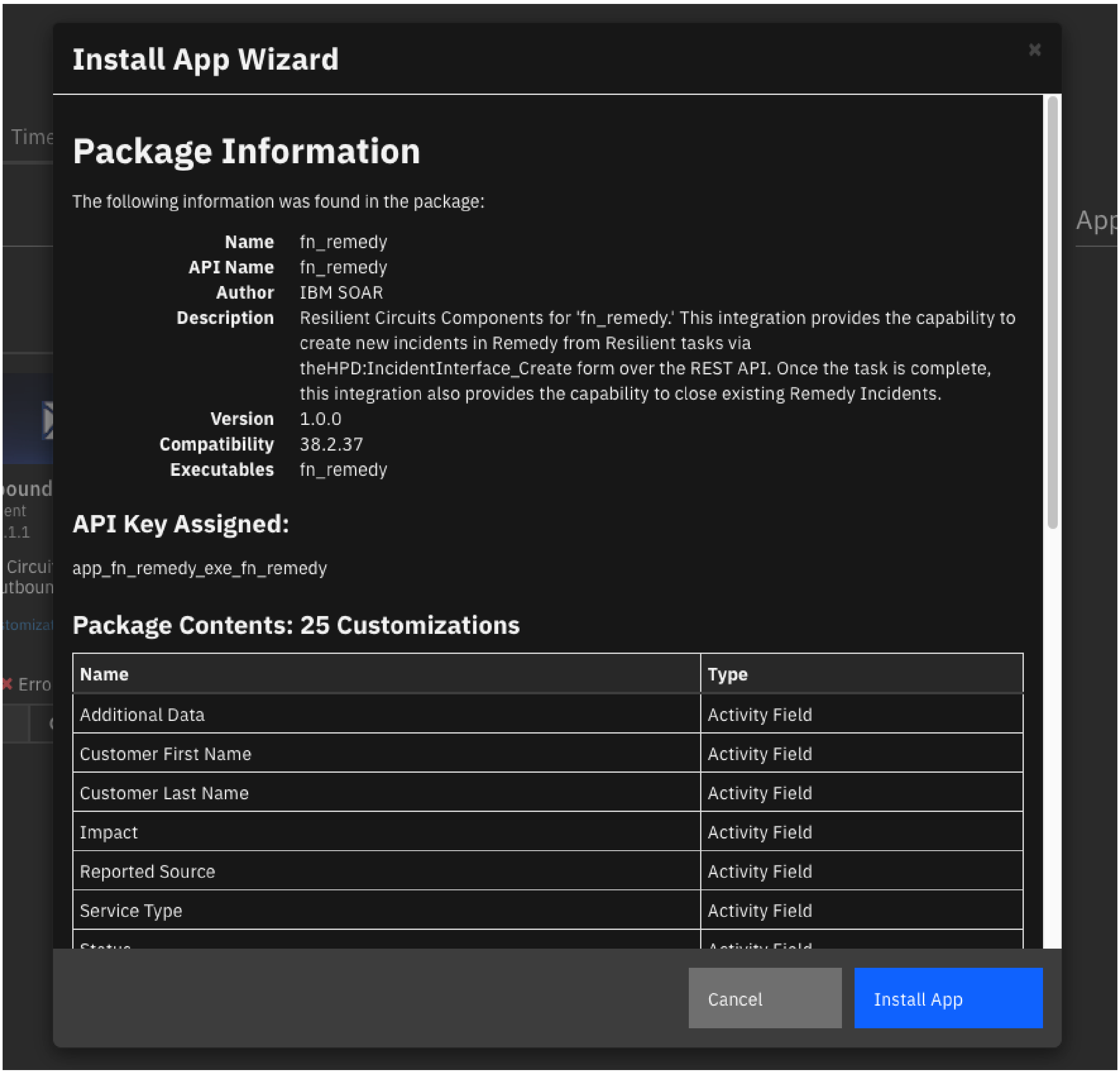
App Configuration

The following table provides the settings you need to configure the app. These settings are made in the app.config file. See the documentation discussed in the Requirements section for the procedure.

Config	Required	Example	Description
remedy_host	Yes	`<example.domain>`	*Hostname for the Remedy instance*
remedy_user	Yes	`<example_user>`	*Username to use to authenticate with Remedy.*
remedy_password	Yes	`xxx`	*Password to use to authenticate with Remedy.*
max_datatable_rows	No	`30`	*Max number of datatable rows to return from the Reilient API when closing an Incident.*
remedy_port	No	`8443`	*Port number where the Remedy REST API is exposed.*
verifye	No	`true`	*Set to `true` to make verified request to Remedy, `false` otherwise.*
http_proxy	No	`example.domain`	*http proxy for request traffic*
http_proxy	No	`example.domain`	*https proxy for request traffic*

Custom Layouts

- Import the Data Tables and Custom Fields like the screenshot below:



Install App Wizard

Package Information

The following information was found in the package:

Name	fn_remedy
API Name	fn_remedy
Author	IBM SOAR
Description	Resilient Circuits Components for 'fn_remedy.' This integration provides the capability to create new incidents in Remedy from Resilient tasks via theHPD:IncidentInterface_Create form over the REST API. Once the task is complete, this integration also provides the capability to close existing Remedy Incidents.
Version	1.0.0
Compatibility	38.2.37
Executables	fn_remedy

API Key Assigned:

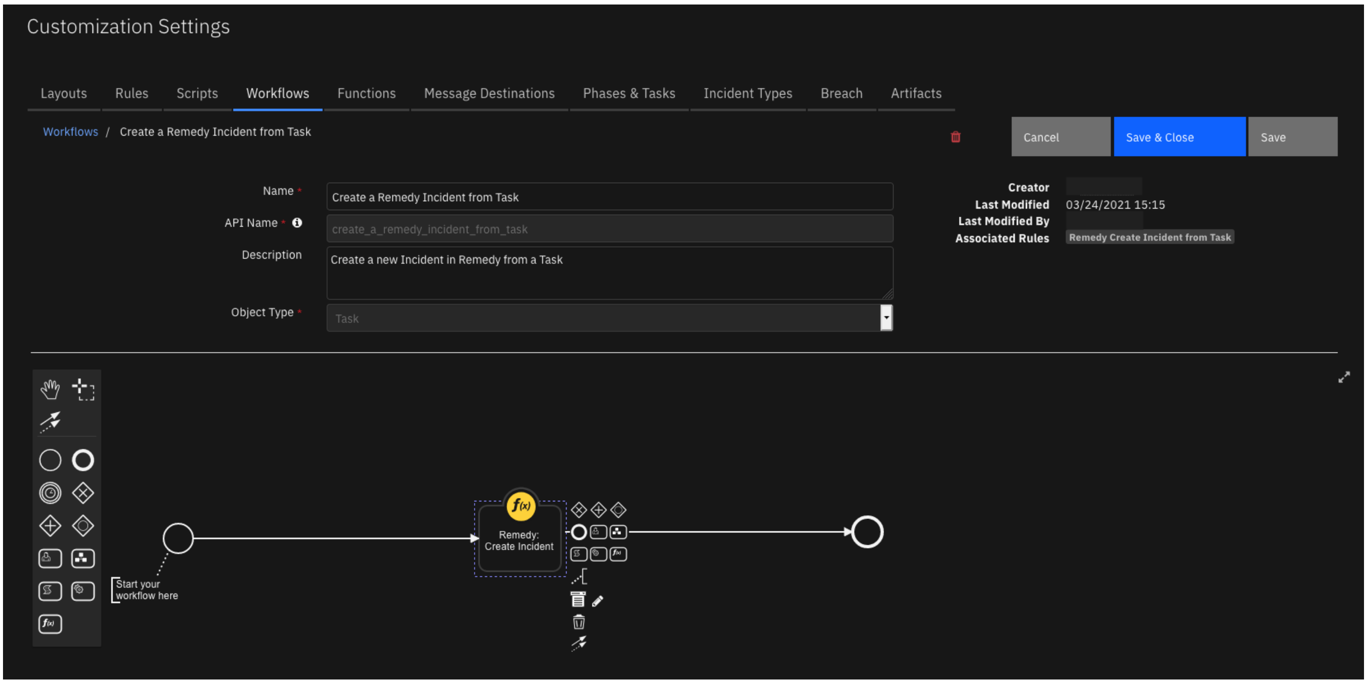
app_fn_remedy_exe_fn_remedy

Package Contents: 25 Customizations

Name	Type
Additional Data	Activity Field
Customer First Name	Activity Field
Customer Last Name	Activity Field
Impact	Activity Field
Reported Source	Activity Field
Service Type	Activity Field
Status	Activity Field

Function - Remedy: Create Incident

Create a new incident in Remedy from a Resilient task.



Name * Create a Remedy Incident from Task

API Name * create_a_remedy_incident_from_task

Description Create a new Incident in Remedy from a Task

Object Type * Task

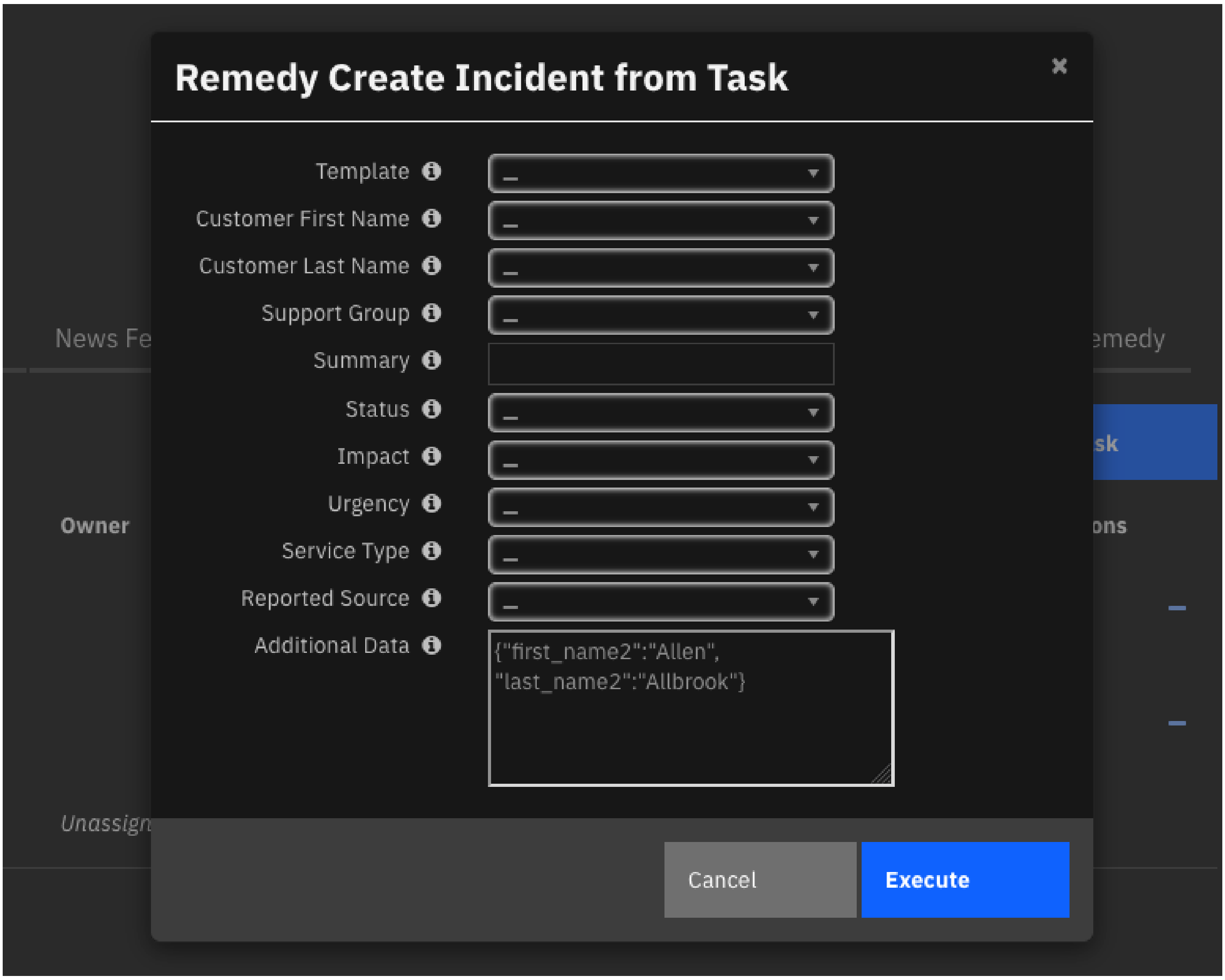
Creator

Last Modified 03/24/2021 15:15

Last Modified By

Associated Rules Remedy Create Incident from Task

Activity Fields



Template —

Customer First Name —

Customer Last Name —

Support Group —

Summary

Status —

Impact —

Urgency —

Service Type —

Reported Source —

Additional Data {"first_name2":"Allen", "last_name2":"Allbrook"}

Remedy is a highly customizable product, and this integration was designed with those customizations in mind.

Note that when creating an incident in Remedy via the REST API, any auto-routing that is configured in the Remedy platform will continue to apply as it would when creating a new incident. This can result in a discrepancy between the data that was submitted and the data that is present in Remedy once the incident object is actually created. For example, the payload sent to Remedy could indicate a Status of New for an incident (either directly or via a template.) However, when that ticket is actually created, the auto-routing in Remedy could be configured to assign it to a user and update the Status to Assigned. This is expected, and the true status of the created incident will be reflected in the datatable.

Templating

To facilitate the use of templates, none of the activity fields are required. If your Remedy server has a template defined that provides all required fields to create an incident, you may simply provided the template name and run the function. Note that it is necessary to manually enter the template name(s) so that they are available in the dropdown. We have provided a stock, out-of-the-box template name as an example.

Other Common Fields

For convenience, several activity fields have been created to handle input for commonly used fields in Remedy such as Status, Impact, and Urgency. These activity fields are not required, as templates can also provide those values. Note that if a template and activity field provide the same value, the activity field will take precedence over the template.

Please note that the user has the ability to customize what values appear in the dropdown menu for each activity field. This action will likely be necessary if not taking advantage of the Remedy's templating functionality via this integration.

Additional Data

Finally, the Additional Data activity field allows the mapping of any other values to the Remedy form not covered in the above activity fields, including custom defined fields. The fields must be provided as a Python-like dictionary. For example:

```
{"Short Description": "example incident text", "my_custom_field": 1, }
```

The keys provided in this dictionary string must match the API names of fields in the `HPD:IncidentInterface` form. To retrieve the schema for this form on the Remedy server, send an HTTP OPTIONS request to `http://serverName/api/arsys/v1/entry/HPD:Interface_Create`. This is the endpoint used to create Remedy incidents over the API, and thus the response will indicate which fields are available to map and which values are acceptable.

▶ Inputs:

Name	Type	Required	Example	Tooltip
`incident_id`	`number`	No	`-`	-
`remedy_incident_name`	`text`	No	`-`	-
`remedy_payload`	`textarea`	No	`-`	-

Name	Type	Required	Example	Tooltip
`task_id`	`number`	No	`-`	-

▶ Outputs:

```
results = {
    "version": "1.0",
    "success": True,
    "reason": None,
    "content": {
        "values": {
            "Incident Number": "INC000000000705",
            "Request ID": "000000000000605",
        },
        "_links": {
            "self": [
                {
                    "href":
"http://35.153.129.209:8008/api/arsys/v1/entry/HPD:IncidentInterface_Creat
e"
                }
            ]
        },
        "task": {
            "name": "Investigate Exposure of Personal Information/Data",
            "inc_id": 2167,
            "inc_owner_id": 1,
            "due_date": None,
            "required": True,
            "owner_id": None,
            "user_notes": None,
            "status": "O",
            "frozen": False,
            "owner_fname": None,
            "owner_lname": None,
            "init_date": 1617903226947,
            "active": True,
            "src_name": None,
            "inc_name": "new remedy incident",
            "instr_text": None,
            "instructions": None,
            "form": "data_compromised, determined_date",
            "members": None,
            "perms": {
                "read": True,
                "write": True,
                "comment": True,
                "assign": True,
                "close": True,
                "change_members": True,
                "attach_file": True,
                "read_attachments": True,
                "delete_attachments": True,
                "change_header": False,
            },
            "notes": [],
            "closed_date": None,
            "actions": [
                {"id": 157, "name": "Remedy Create Incident from Task",
"enabled": True}
            ],
            "phase_id": 1005,
            "category_id": 1,
            "notes_count": 0,
            "attachments_count": 0,
            "task_layout": [],
            "auto_deactivate": True,
            "creator_principal": {
                "id": 1,
                "type": "user",
                "name": "a@example.com",
                "display_name": "Resilient Administrator",
            },
            "regs": {"88": "Data Breach Best Practices"},
            "custom": False,
            "id": 378,
            "inc_training": False,
            "cat_name": "Respond",
            "description": "",
            "at_id": None,
            "private": None,
        },
    },
    "raw": '{"values": {"Incident Number": "INC000000000705", "Request
ID": "000000000000605"}, "_links": {"self": [{"href":
"http://35.153.129.209:8008/api/arsys/v1/entry/HPD:IncidentInterface_Creat
e"}]}}',
    "inputs": {
        "incident_id": 2167,
        "remedy_incident_name": "Investigate Exposure of Personal
Information/Data",
        "remedy_payload": {
            "format": "text",
            "content": '{"ApplyTemplate": "Email Issue", "First_Name":
"Allen", "Last_Name": "Allbrook", "Impact": "1-Extensive/Widespread",
"Urgency": "1-Critical", "Service_Type": "User Service Restoration",
"Status": "New", "Reported Source": "Direct Input", "Description": null,
"Assigned Support Organization": "Service Desk", "additional_data":
{"format": "text", "content": null}}',
        },
        "task_id": 378,
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-remedy",
        "package_version": "1.0.0",
        "host": "example.host.net",
        "execution_time_ms": 4199,
        "timestamp": "2021-04-08 13:34:11",
    },
}
```

▶ Example Pre-Process Script:

```
# Importing JSON means this function has a hard requirement on the python 3 feature.
import json

# Any of the selected Activity Fields in the rule are taken in and formed as a dict
payload = {
  "ApplyTemplate": rule.properties.remedy_template,
  "First_Name": rule.properties.remedy_first_name,
  "Last_Name": rule.properties.remedy_last_name,
  "Impact": rule.properties.remedy_impact,
  "Urgency": rule.properties.remedy_urgency,
  "Service_Type": rule.properties.remedy_service_type,
  "Status": rule.properties.remedy_status,
  "Reported Source": rule.properties.remedy_reported_source,
  "Description": rule.properties.remedy_note,
  "Assigned Support Organization": rule.properties.remedy_support_group,
  "additional_data": rule.properties.remedy_additional_data
}

# set inputs
inputs.task_id = task.id
inputs.incident_id = incident.id
inputs.remedy_incident_name = task.name
inputs.remedy_payload = json.dumps(payload)
```

▶ Example Post-Process Script:

```
noteText = "<h5> Remedy Create Incident</h5>"

if results["success"]:
  noteText += "<p>Successfully sent task {0} \"{1}\" to Remedy as Incident Number {2} (UI name) and Request ID {3} (API name).</p>"\
  "".format(results["content"]["task"]["id"], results["content"]["task"]["name"],\
  results["content"]["values"]["Incident Number"], results["content"]["values"]["Request ID"])
else:
  noteText += "<p>Unable to send task {0} \"{1}\" to
```

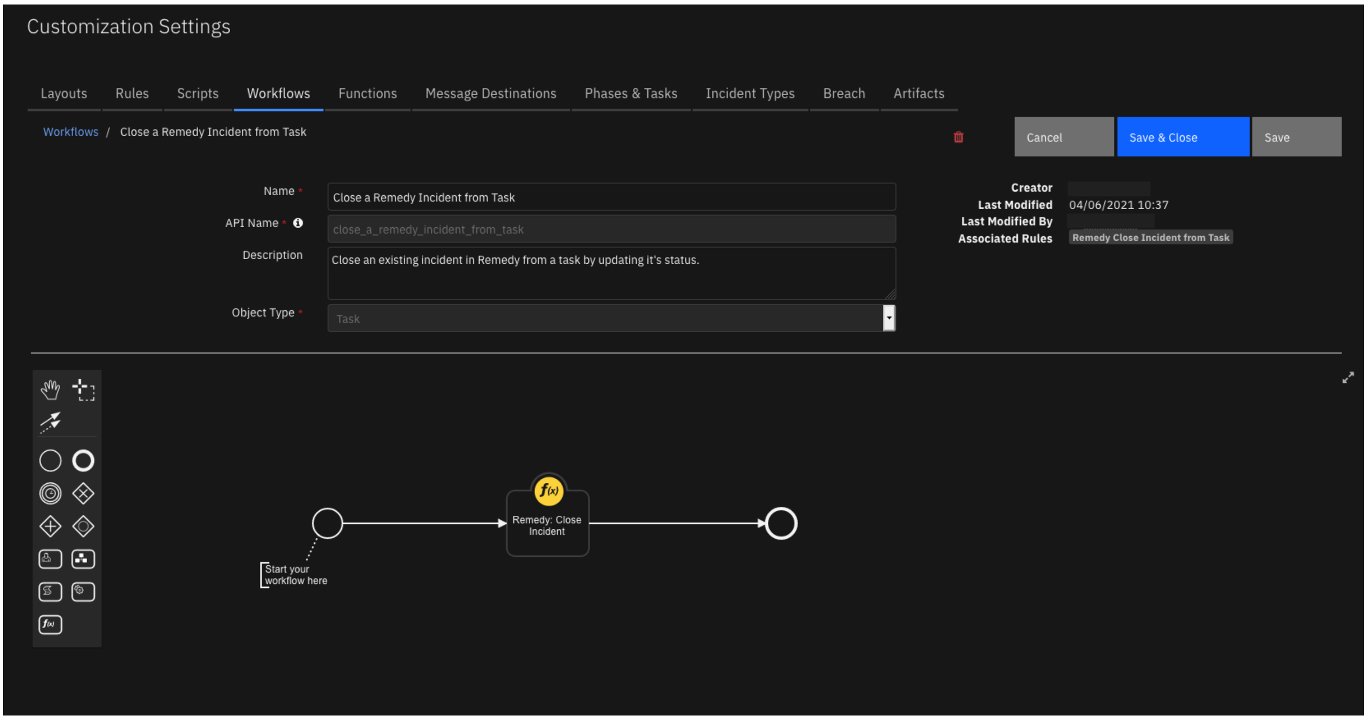
```
Remedy</p>".format(results["content"]["task"]["id"], results["content"]
["task"]["name"])
  noteText += "<p>Ensure the activity fields and payload you provide meet
the minimum requirements in your system for incident creation and
routing."

richText = helper.createRichText(noteText)
incident.addNote(richText)
```

Function - Remedy: Close Incident

Close an incident ticket in Remedy by modifying its status. The function will make an API call to Remedy to retrieve the target incident form. If the status of that form is "Resolved", "Closed", or "Cancelled," no change to the incident is made. Otherwise, the status is updated to Resolved with Status Reason "No Further Action Required" and Resolution "Closed from CP4S."

When a task is closed under a case, an automatic rule will trigger containing this function. If a row in the Remedy datatable matches the name and ID of the task just closed, the above logic will trigger to ensure that the corresponding incident in Remedy is also closed.

In the event that multiple rows in the datatable match the target task (a task has been raised to Remedy more than once), the function will iterate over those rows to ensure that each of the corresponding incidents in Remedy are closed.



Name * Close a Remedy Incident from Task

API Name * close_a_remedy_incident_from_task

Description Close an existing incident in Remedy from a task by updating it's status.

Object Type * Task

Creator [redacted]

Last Modified 04/06/2021 10:37

Last Modified By [redacted]

Associated Rules Remedy Close Incident from Task

▶ Inputs:

Name	Type	Required	Example	Tooltip
`incident_id`	`number`	No	`-`	-
`remedy_payload`	`textarea`	No	`-`	-

Name	Type	Required	Example	Tooltip
`task_id`	`number`	No	`–`	-

▶ Outputs:

```
results = {
    "version": "1.0",
    "success": True,
    "reason": None,
    "content": {"closed": ["INC000000000807|INC000000000807"], "skipped":
[]},
    "raw": '{"closed": ["INC000000000807|INC000000000807"], "skipped":
[]}',
    "inputs": {
        "incident_id": 2167,
        "remedy_payload": {"format": "text", "content": '{"Status_Reason":
"foo"}'},
        "task_id": 378,
    },
    "metrics": {
        "version": "1.0",
        "package": "fn-remedy",
        "package_version": "1.0.0",
        "host": "example.host.net",
        "execution_time_ms": 16170,
        "timestamp": "2021-04-08 13:42:52",
    },
}
```

▶ Example Pre-Process Script:

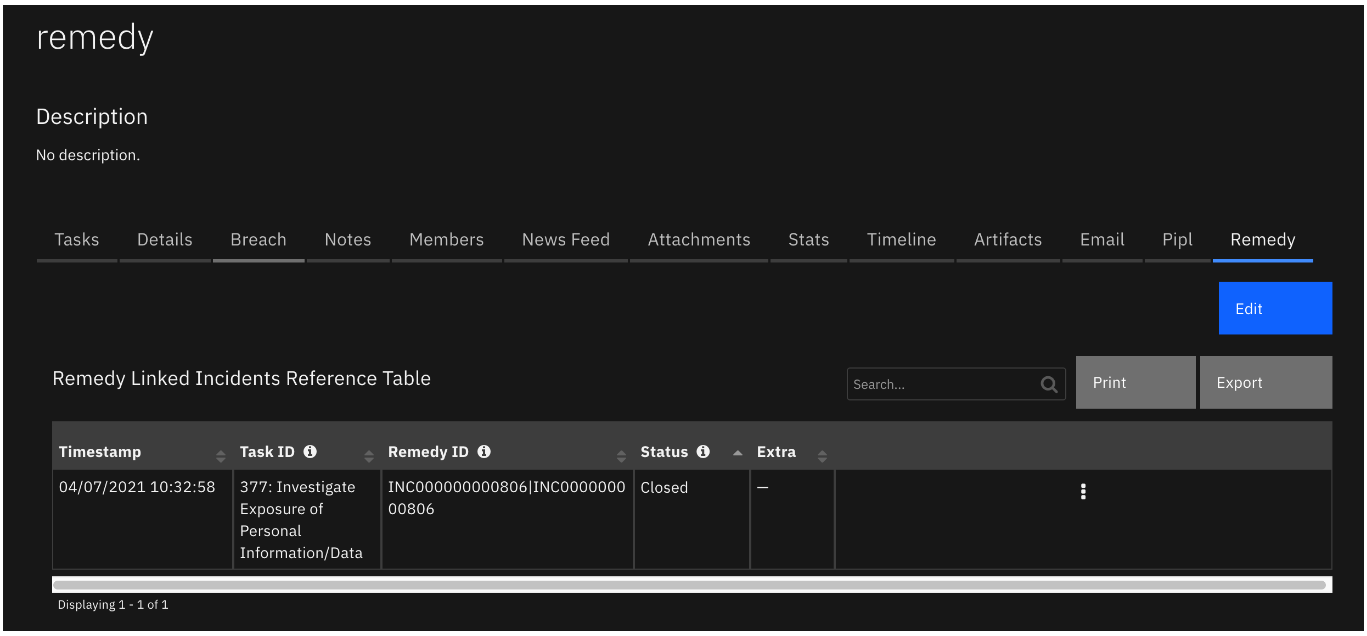
```
# Importing JSON means this function has a hard requirement on the python
3 feature.
import json

inputs.task_id = task.id
inputs.incident_id = incident.id

payload = {}

# Use this section to add key, value pairs to send to Remedy
# These values will be added/updated on the target Remedy incident,
# so they must conform with the "HPD:IncidentInterface_Create" schema

payload["Status_Reason"] = "foo"
# payload["policy_name"] = "bar"
```

```
inputs.remedy_payload = json.dumps(payload) if payload else ''
```

▶ Example Post-Process Script:

```
noteText = "<h5>Remedy Close Incident:</h5>"

if results["success"]:
  if not results["content"]["closed"] and not results["content"]
["skipped"]:
    "<p>No record of task ID {0} in Remedy. Nothing was updated.
</p>".format(task.id)
  else:
    if results["content"]["closed"]:
      noteText += "<p>The following incidents were matched in Remedy and
the successfully closed:</p>"
      for item in results["content"]["closed"]:
        noteText += "<p>    Incident Number {0}, Request ID: {1}
</p>".format(item["values"]["Incident Number"], item["values"]["Request
ID"])
    if results["content"]["skipped"]:
      noteText += "<p>The following incidents were not able to be closed.
Common reasons include that the incident has been previously closed, " \
      "the incident has been deleted, or the payload sent to Remedy was
incomplete according to the requirements of your specific system:</p>"
      for item in results["content"]["skipped"]:
        noteText += "<p>    Incident Number {0}, Request ID: {1}
</p>".format(item["values"]["Incident Number"], item["values"]["Request
ID"])
else:
  noteText += "<p>Function failed to complete.</p>"

richText = helper.createRichText(noteText)
incident.addNote(richText)
```

Data Table - Remedy Linked Incidents Reference Table



Timestamp	Task ID	Remedy ID	Status	Extra
04/07/2021 10:32:58	377: Investigate Exposure of Personal Information/Data	INC000000000806|INC000000000806	Closed	—

API Name:

remedy_linked_incidents_reference_table

Columns:

Column Name	API Access Name	Type	Tooltip
Extra	`extra`	`textarea`	Incident ID of the Remedy form entry
Remedy ID	`remedy_id`	`text`	Request ID of the Remedy form entry
Status	`status`	`textarea`	Last status applied to the Remedy Incident
Task ID	`taskincident_id`	`text`	ID of the Task and its description
Timestamp	`timestamp`	`datetimepicker`	-

Data explanation

The Remedy Linked Incidents Reference Table will be updated when either the `Remedy: Create Incident` or `Remedy: Close Incident` functions complete.

Remedy: Create Incident

Once an incident is posted to Remedy, the auto-routing feature has the potential to alter the values of some of the fields within the Remedy incident (see Activity Fields for more information on this). Due to this potential, the integration will fetch the incident back from the Remedy server after it has been created to ensure the datatable is updated with accurate information. Once the incident data is received from Remedy, relevant fields are recorded in the datatable.

One item to note is the difference between the `Remedy ID` and `Extra` columns. As noted in columns, above, both of these columns contain some sort of ID value relevant to the Remedy Incident.

The `Remedy ID` column contains the "Request ID" of the form entry. Often, this value is of the form `INCxxxxxxx|INCxxxxxxx`. Although this notation may appear to be two numbers separated by the `|` character, the entire string together is a single Request ID value. This ID is used to refer to the incident over the API.

The `Extra` columns contains the "Incident Number" of the form entry. Often this value is of the form `INCxxxxx`, but will likely not look like either component of the Request ID. This Incident Number is the ID that appears in the UI for the incident inside the Incident Management Console.

Remedy: Close Incident

Once an Incident is Closed in Remedy, the datatable will be updated with the new status of that incident.

Rules

Rule Name	Object	Workflow Triggered
Remedy Create Incident from Task	task	`create_a_remedy_incident_from_task`
Remedy Close Incident from Task	task	`close_a_remedy_incident_from_task`

Troubleshooting & Support

Refer to the documentation listed in the Requirements section for troubleshooting information.

For Support

This is an IBM supported app. Please search https://ibm.com/mysupport for assistance.