README.md

User Guide: Microsoft Exchange Online Functions for IBM Resilient v1.0.0

Table of Contents

Key Features

Resilient Integration with Exchange Online provides the capability to access and manipulate Microsoft Exchange Online (Office 365 in the cloud) messages from the IBM Resilient Soar Platform. The integration uses Microsoft Graph API to access the data in Office 365. Included in the integrations are the following capabilities:

- Get the user profile of the specified email address in JSON format.
- Get a specified message and and return the results in JSON format.
- Get a specified message in .eml format and write as an incident attachment.
- Move a message to a specified "Well-known" Outlook folder.
- Send an message: from the specified email address to the specified recipients with specified message subject and body text.
- Query messages of a single user, a list of users, or the whole tenant and return a list of messages matching the criteria: message sender, messages from a specific Well-known folder, a time frame for when the message was received, text contained in the message subject or the message body, whether the message has attachments. Results are returned in the Exchange Online Query Message Results data table.
- Delete a single specified message from a specified email address.
- Delete a list of messages that are the results of a message query. The messages deleted are written to the Exchange Online Query Messages data table.
- Create a meeting event in the organizer's Outlook calendar and send a calendar event message to meeting participants inviting them to the meeting.

Integration Flow for Phishing Investigation Use Case

The Exchange Online integration primary use case is to monitor and control email activities in Exchange Online (Office 365 Outlook in the cloud) and protect against inbound malicious emails.

To use the integration, run the Query Messages rule from the Action menu of an incident. From the rule, you can search a single email address, a list of email addresses or the entire tenant. Results of the query are returned in the Exchange Online Message Query Results data table on the Exchange Online incident tab. Each row in the data table contains information from one message and the following actions can be performed on each message when it's state is Active in the Status column:

- Create artifacts: Email Recipient, Email Sender, Email Subject.
- Delete the message.
- Move the message to a Well-known folder.
- Write the message .eml as an incident.
- Write the message JSON returned from MS Graph to an incident note.

The data table Status column is set to Active when the message is entered in the table. Any time after that, a user can delete the message; however, this could update the Status field to Not Found or Deleted if the message is deleted when running one of the above data table rules or workflows.

The first column of the data table displays the time the query occurred. You can use this value to sort through data if multiple queries are run and entered into the data table. You may want to empty the data table after each query.

Because a large number of messages can be returned from a query, the integration has following parameters in the app.config to limit the number of messages returned:

- max_messages
- max_users

Considering using these parameters to improve performance when running queries.

You can perform additional investigation, including using other email analysis scripts and integrations, on messages after writing the message to a note or attachment.

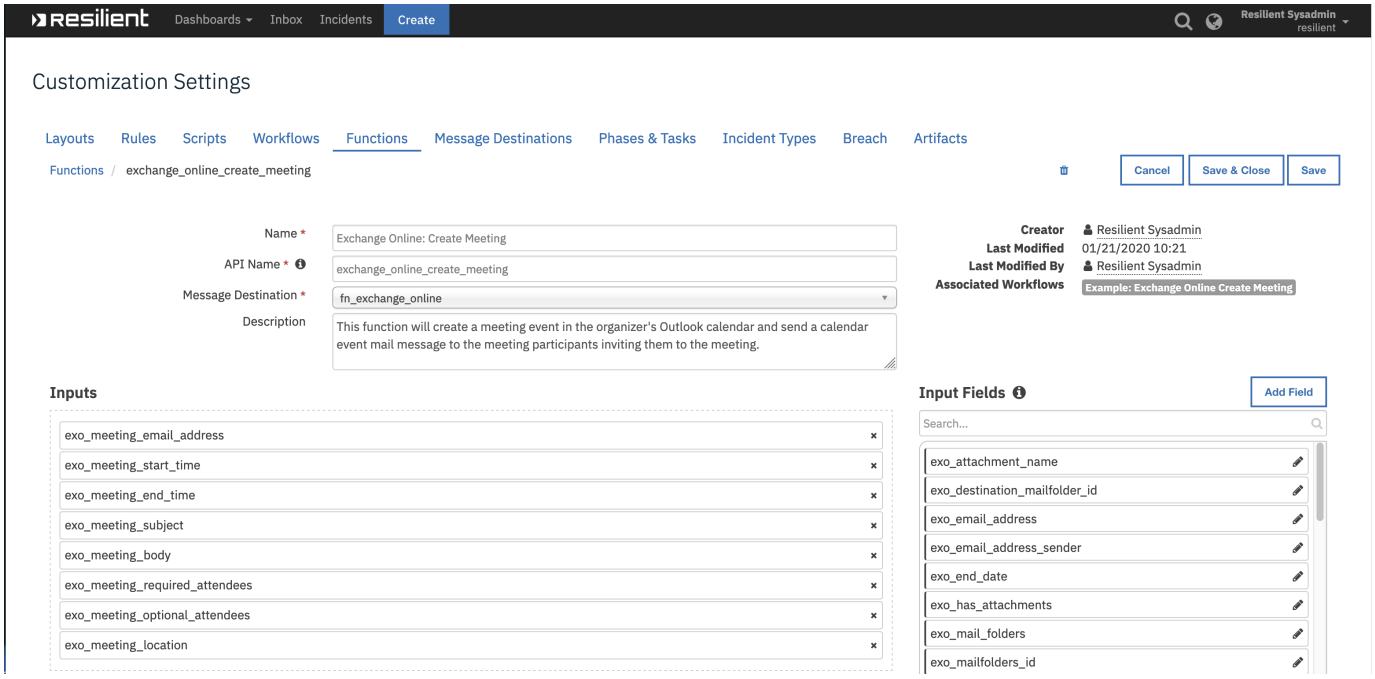
Once you complete the investigation of the messages and there are problematic messages that you want to delete, use the Example: Exchange Online Delete Message rule from the incident’s Actions menu. The rule starts a workflow that performs a query of messages and sends the matching results to a function that deletes a list of messages. The results are written to the Exchange Online Message Query Results data table with a Status column of "Deleted" in red. An incident note is also written that indicates the number of messages deleted.

Use this rule with caution as you can delete many user messages.

At anytime the user can send a message or schedule a meeting using the Exchange Online: Send Message and Exchange Online: Create meeting rules and workflows.

Function - Exchange Online: Create Meeting

The Exchange Online: Create Meeting function creates a meeting event in the organizer's Outlook calendar and sends a calendar event invitation message to the meeting participants.



▼ Inputs:

Name	Type	Required	Example	Tooltip
exo_meeting_body	text	Yes	meeting message body	Meeting message body
exo_meeting_email_address	text	Yes	user@example.com	Email address of meeting coordinator
exo_meeting_end_time	datetimepicker	Yes	-	End date and time for meeting
exo_meeting_location	text	No	-	-
exo_meeting_optional_attendees	text	No	user1@example.com, user2@example.com	Comma separated list of optional attendee email addresses
exo_meeting_required_attendees	text	No	user1@example.com, user2@example.com	Comma separated list of required attendee email addresses
exo_meeting_start_time	datetimepicker	Yes	-	Meeting start date and time
exo_meeting_subject	text	Yes	-	Meeting Subject

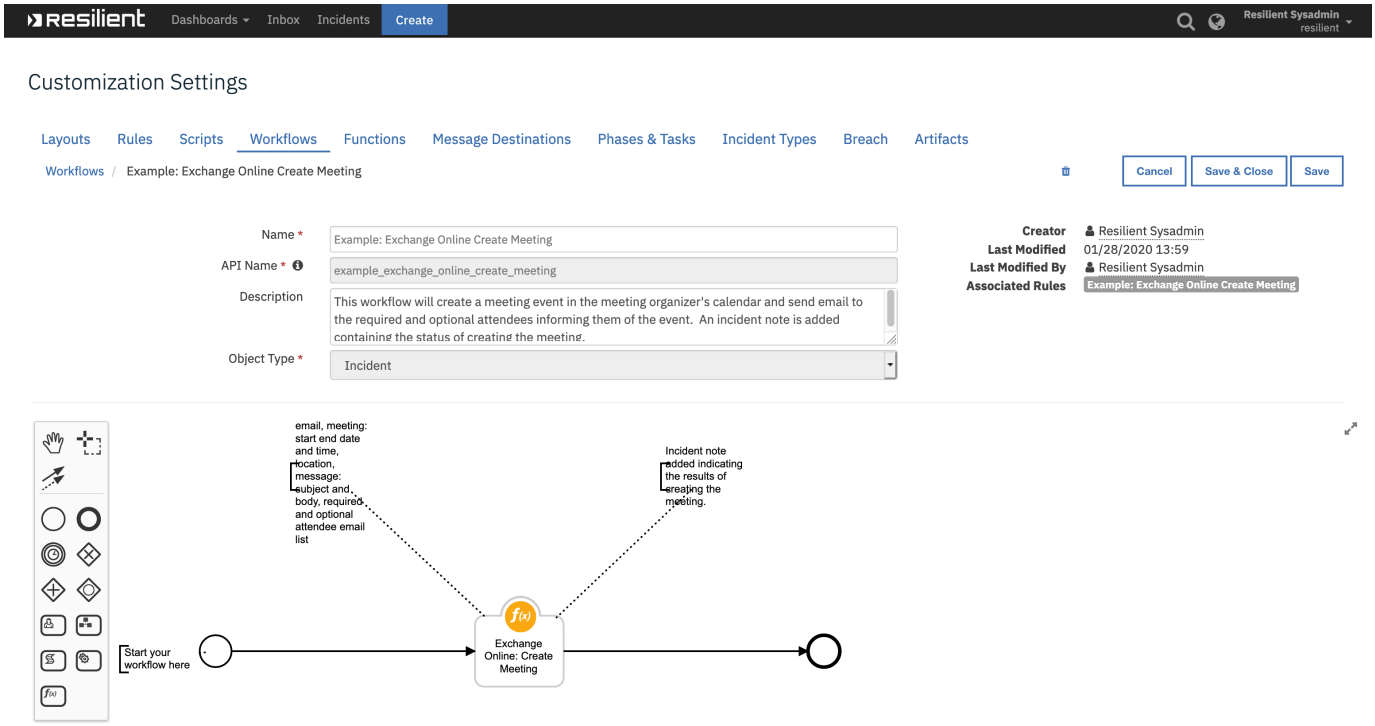
▼ Outputs:

```
results = {
    'inputs': {
          u'exo_meeting_end_time': 1581022800000,
          u'exo_meeting_optional_attendees': None,
          u'exo_meeting_subject': u'phishing meeting',
          u'exo_meeting_body': u'<div class="rte"><div>We need to talk about this!</div></div>', u'exo_meeting_require
          u'exo_meeting_start_time': 1581004800000,
          u'exo_meeting_email_address': u'resilient2@securitypocdemos.onmicrosoft.com', u'exo_meeting_location': None}
    'metrics': {'package': 'fn-exchange-online',
```

```
        'timestamp': '2020-02-04 13:30:45',
        'package_version': '1.0.0',
        'host': 'MacBook-Pro.local',
        'version': '1.0',
        'execution_time_ms': 1728},
    'success': True,
      'content': {u'body': {u'content': u'', u'contentType': u'html'}, u'sensitivity': u'normal', u'locations': [], ..
    'reason': None,
    'version': '1.0',
    'pretty_string': u'{\n    "body": {\n        "content": "",\n        "contentType": "html"\n    },\n    "sensitivi
    'content': {u'body': {u'content': u'', u'contentType': u'html'}, u'sensitivity': u'normal', u'locations': [],.....
}
```


▼ Workflows:

The Example: Exchange Online Create Meeting workflow calls the create meeting function and then write the results to an incident note.



▼ Example Pre-Process Script:

```
inputs.exo_meeting_email_address = inputs.exo_meeting_email_address  if rule.properties.exo_meeting_email_address is N
inputs.exo_meeting_start_time = inputs.exo_meeting_start_time if rule.properties.exo_meeting_start_time is None else r
inputs.exo_meeting_end_time = inputs.exo_meeting_end_time if rule.properties.exo_meeting_end_time is None else rule.pr
inputs.exo_meeting_subject = inputs.exo_meeting_subject if rule.properties.exo_meeting_subject is None else rule.prope
inputs.exo_meeting_body = inputs.exo_meeting_body if rule.properties.exo_meeting_body.content is None else rule.proper
inputs.exo_meeting_required_attendees = inputs.exo_meeting_required_attendees if rule.properties.exo_meeting_required_
inputs.exo_meeting_optional_attendees = inputs.exo_meeting_optional_attendees if rule.properties.exo_meeting_optional_
inputs.exo_meeting_location = inputs.exo_meeting_location if rule.properties.exo_meeting_location is None else rule.pr
```

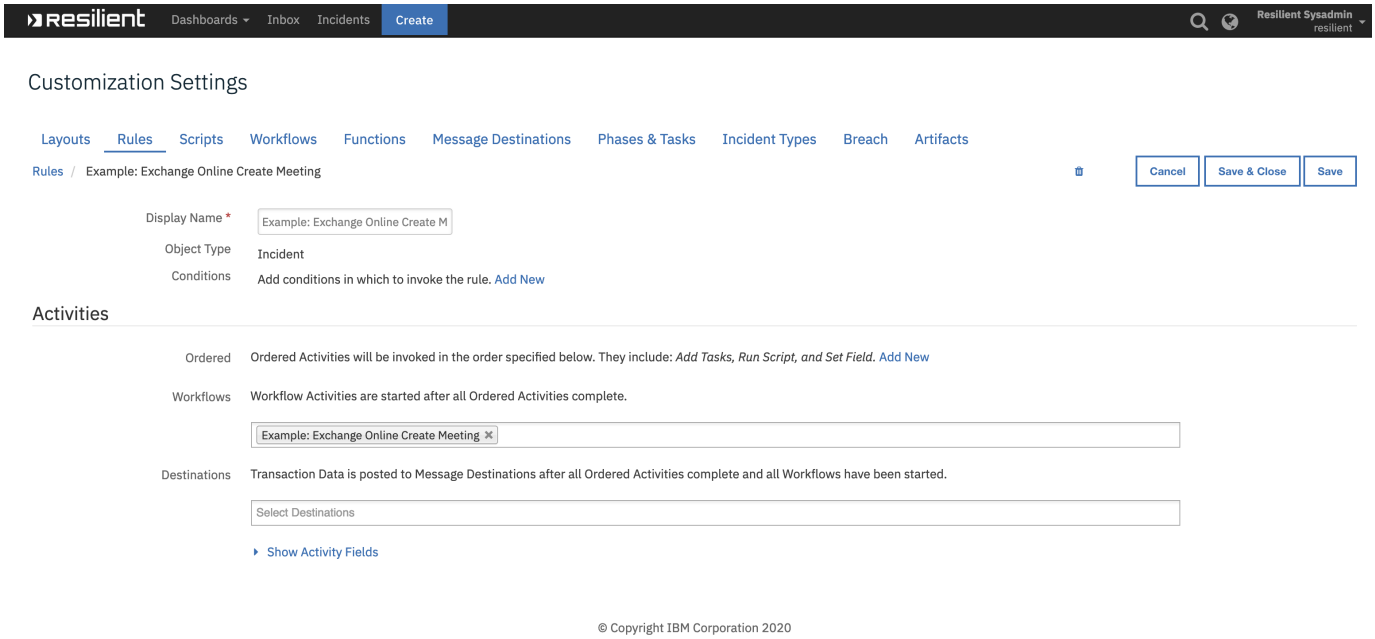
▼ Example Post-Process Script:

```
if results.success:
  noteText = u"Exchange Online created meeting\n   From: {0}\n{1}".format(results.inputs["exo_meeting_email_address"],
else:
  noteText = u"Exchange Online meeting was NOT created\n   From: {0}\n{1}".format(results.inputs["exo_meeting_email_ad
```

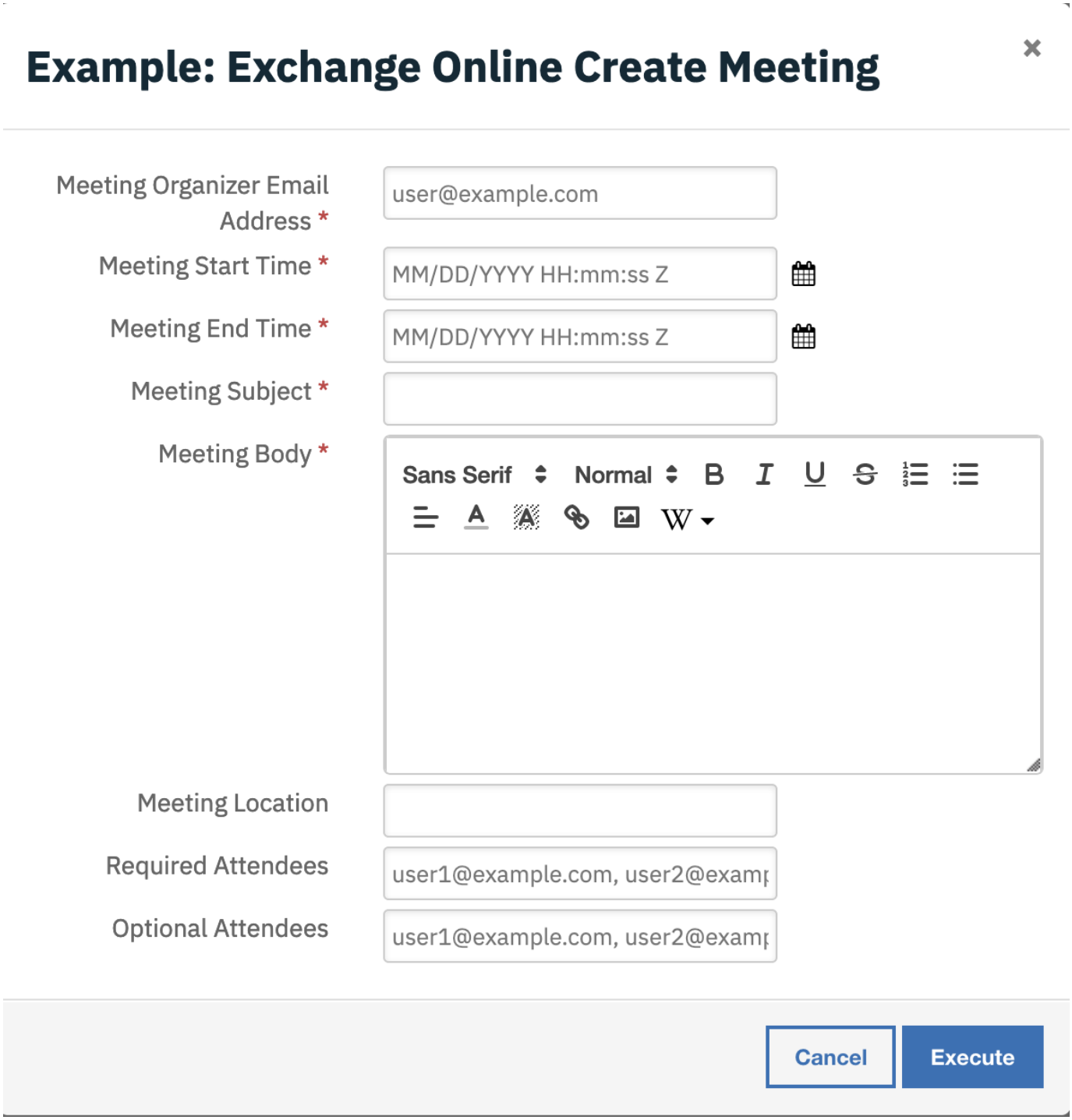
```
incident.addNote(noteText)
```

▼ Example Rule:

The following Example: Exchange Online Create Meeting incident menu item rule is included to create a meeting via Exchange Online:

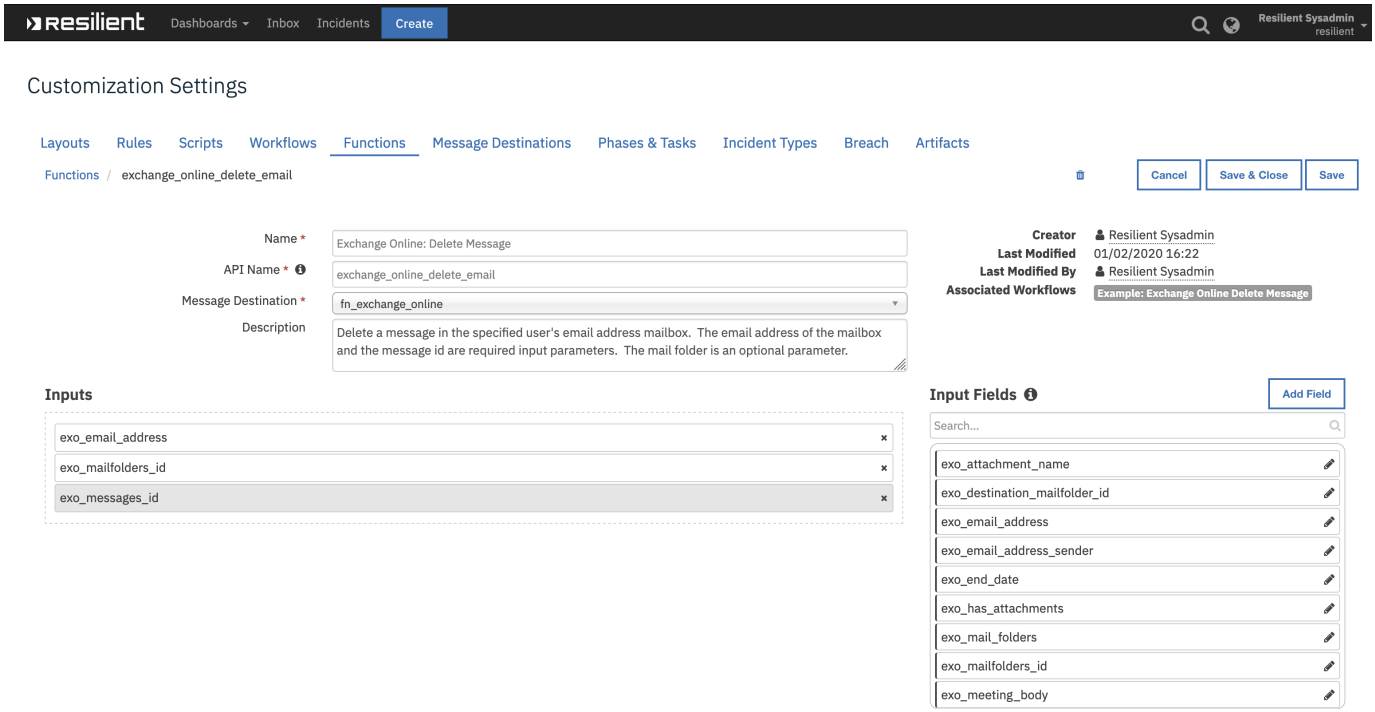


When the Example: Exchange Online Create Meeting rule is activated the following rule activity popup dialog appears prompting for input for creating the meeting and sending a message to the invitees:



Function - Exchange Online: Delete Message

Delete a message in the specified user's email address mailbox. The email address of the mailbox and the message ID are required input parameters. The mail folder is an optional parameter.



▼ Inputs:

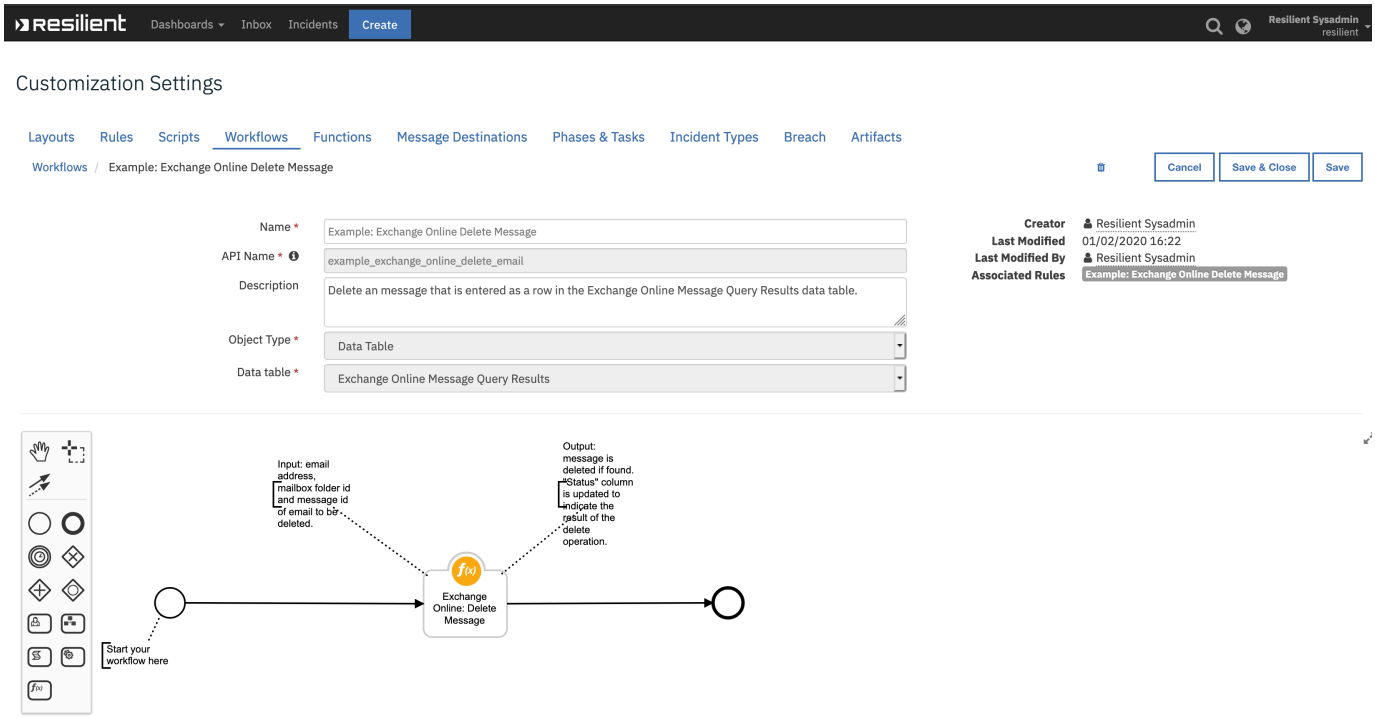
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_mailfolders_id	text	No	inbox	MailFolders ID
exo_messages_id	text	Yes	–	The message ID of the message to be deleted

▼ Outputs:


```
results = {
   'inputs': {u'exo_mailfolders_id': None,
              u'exo_messages_id': u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPu
    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-02-04 09:08:13',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 1300},
    'success': True,
    'content': {'value': True},
    'raw': '{"value": true}',
    'reason': None,
    'version': '1.0'
}
```


▼ Workflows:

The example Delete Message workflow uses the information in the Query Results data table. After the message is deleted using the Example: Exchange Online Delete Message workflow, the "Status" column in the data table is updated from Active to Deleted in red text. Any data table rules accessing the deleted message become inactive.



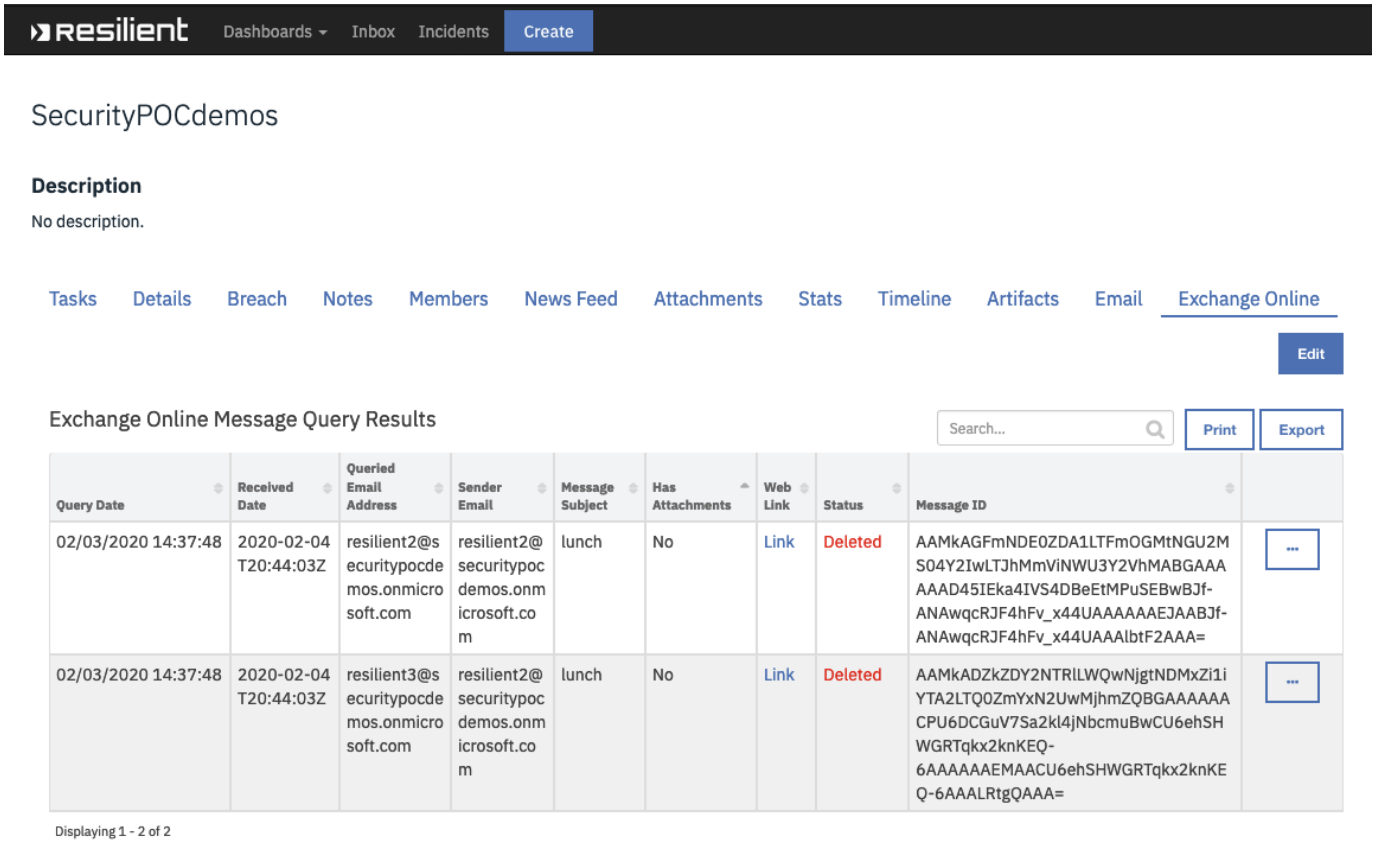
▼ Example Pre-Process Script:

```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
inputs.exo_mailfolders_id = None
```

▼ Example Post-Process Script:

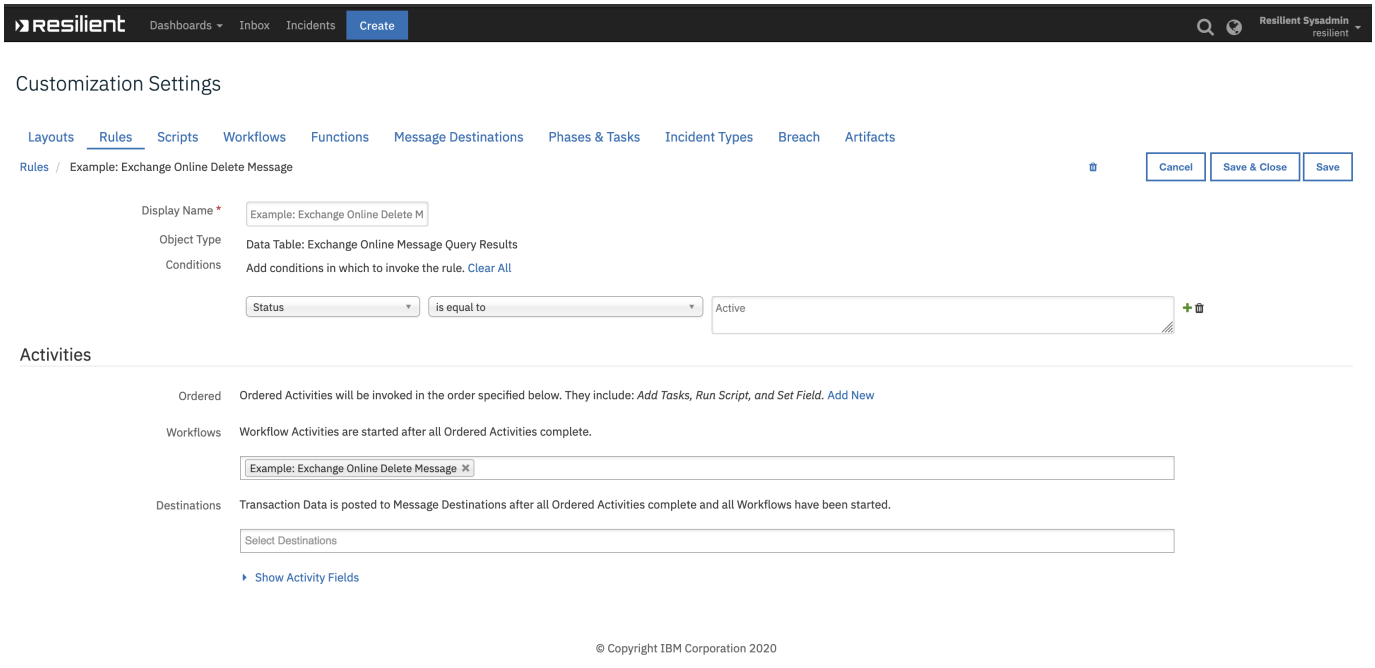
```
if results.success:
  # The message was deleted, so update "status" column in data table.
  text = u"""<p style= "color:{color}">{status} </p>""".format(color="red", status="Deleted")
  row['exo_dt_status'] = helper.createRichText(text)
elif results.content["error"] is not None:
  # There is an "item not found" error mostly likely here
  row['exo_dt_status'] = helper.createRichText(results.content["error"]["code"])
```

▼ Example Workflow Output:



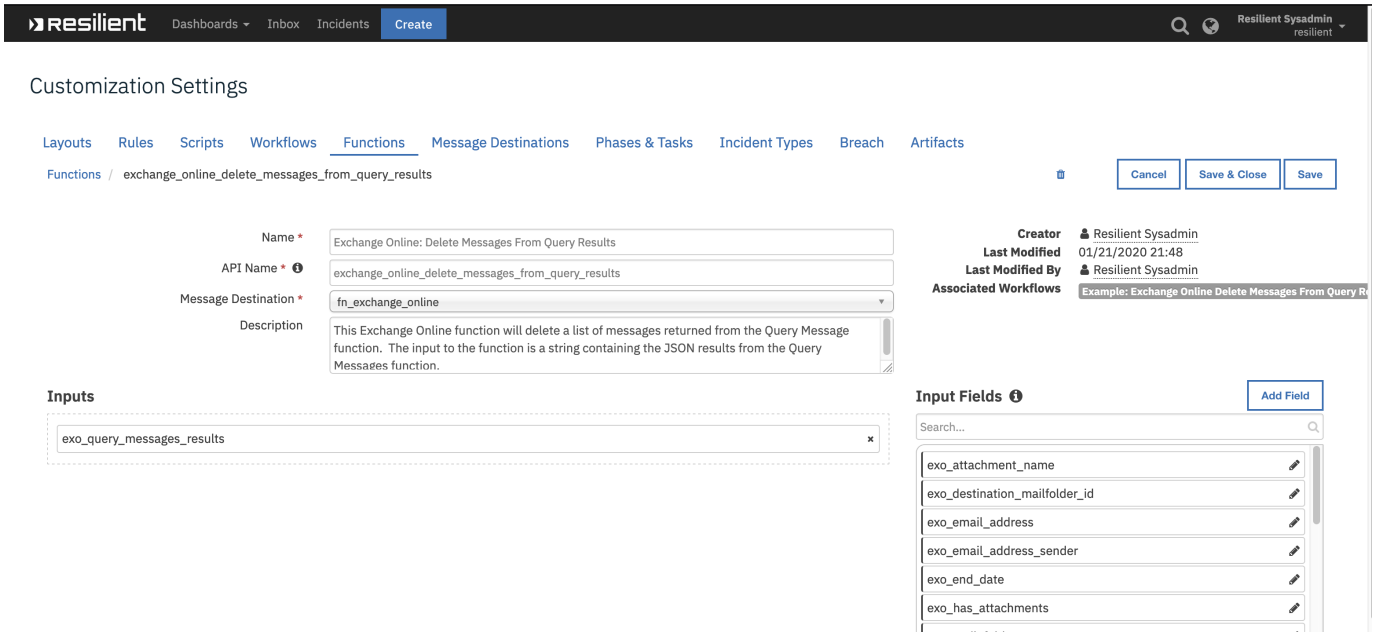
▼ Example Rule:

The Example: Exchange Online Delete Message rule works off the Query Results data table. The Delete Message rule is available when the "Status" column of the message row-entry is set to Active.



Function - Exchange Online: Delete Messages From Query Results

This Exchange Online function deletes a list of messages returned from the Query Message function. The input to the function is a string containing the JSON results from the Query Messages function.



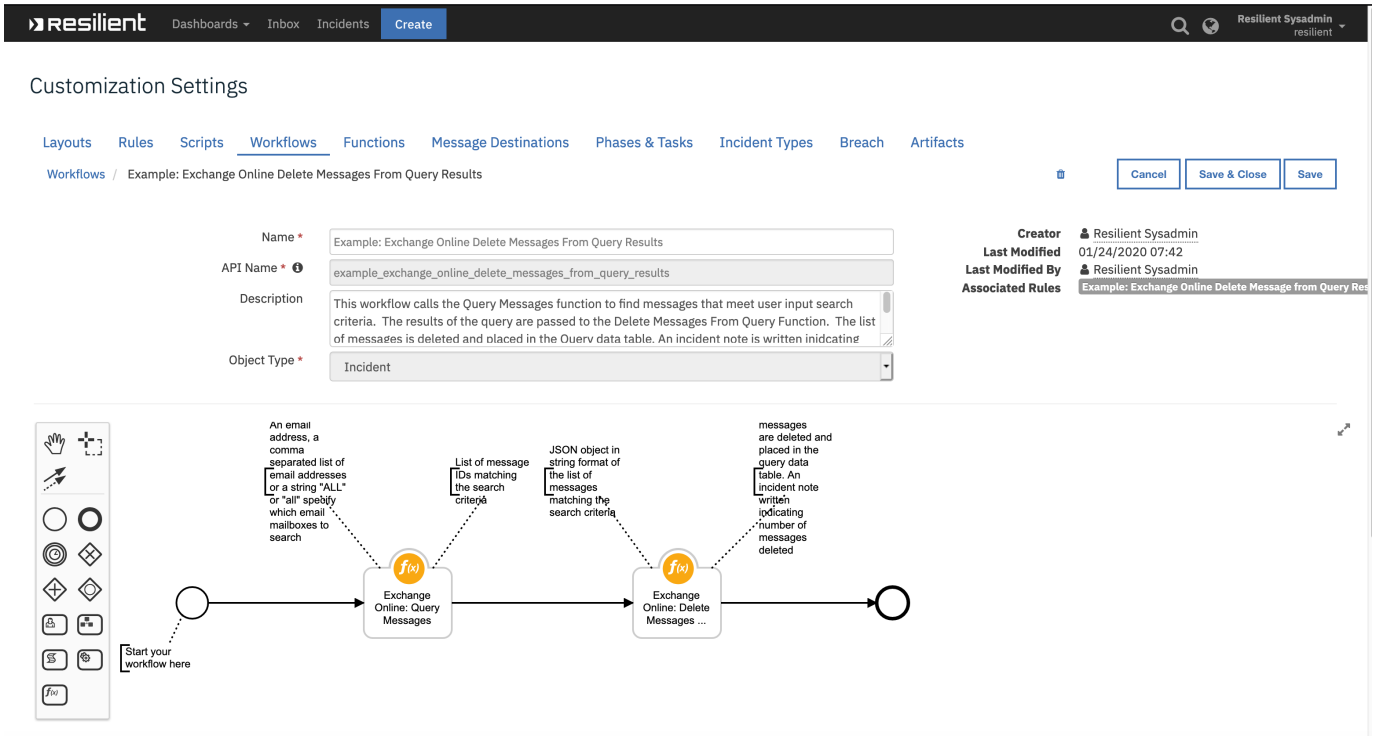
▼ Inputs:

Name	Type	Required	Example	Tooltip
exo_query_messages_results	text	Yes	-	String containing JSON data results from Query Messages function

▼ Outputs:

```
results = {'inputs': {u'exo_query_messages_results': u'[{"status_code": 200, "email_address": "resilient3@securitypocd

'metrics': {'package': 'fn-exchange-online',
            'timestamp': '2020-02-05 15:06:25',
            'package_version': '1.0.0',
            'host': 'MacBook-Pro.local',
            'version': '1.0',
            'execution_time_ms': 4869},

'success': True,
'content': [{'not_deleted_list': [], 'deleted_list': [{u'sentDateTime': u'2020-02-05T20:03:21Z', u'webLink': u'https:/

'reason': None,
'version': '1.0'
}
```

▼ Workflows:



▼ Example Pre-Process Script:

```
inputs.exo_query_messages_results = workflow.properties.exo_query_results['raw']
```

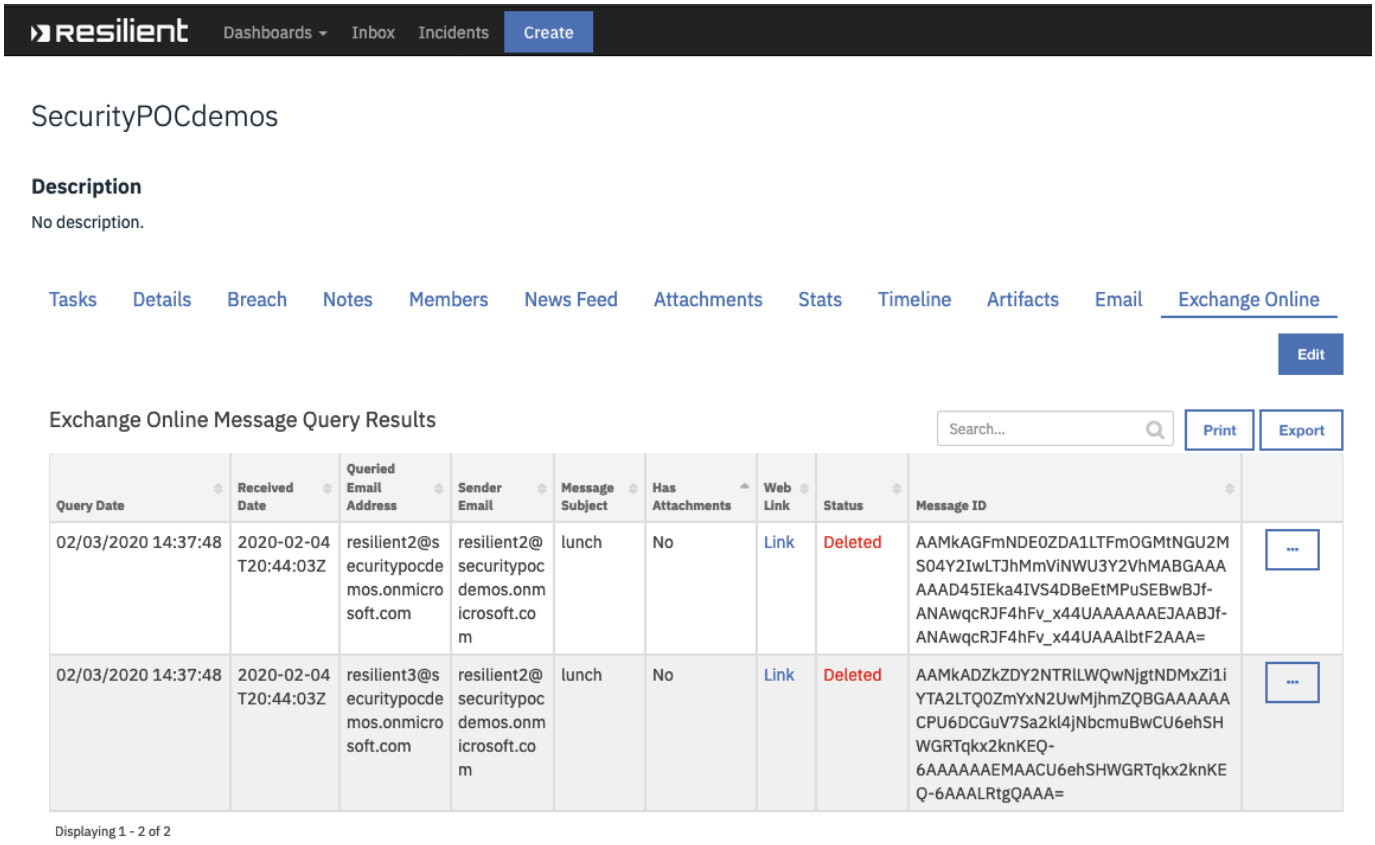
▼ Example Post-Process Script:

```
from java.util import Date

deleted_count = 0
not_deleted_count = 0

# Add each email as a row in the query results data table
for user in results["content"]:

  not_deleted_count = not_deleted_count + len(user["not_deleted_list"])
  for email in user["deleted_list"]:
    deleted_count = deleted_count + 1
    message_row = incident.addRow("exo_message_query_results_dt")
    message_row.exo_dt_query_date = Date()
    message_row.exo_dt_message_id = email.id
    message_row.exo_dt_received_date    = email.receivedDateTime
    message_row.exo_dt_email_address = user["email_address"]
    if email.sender:
      message_row.exo_dt_sender_email = email.sender.emailAddress.address
    else:
      message_row.exo_dt_sender_email = ""
    message_row.exo_dt_message_subject = email.subject
    message_row.exo_dt_has_attachments = email.hasAttachments
    if email.webLink:
      ref_html = u"""<a href='{0}'>Link</a>""".format(email.webLink)
      message_row.exo_dt_web_link = helper.createRichText(ref_html)
    else:
      message_row.exo_dt_web_link = ""

    text = u"""<p style= "color:{color}">{status} </p>""".format(color="red", status="Deleted")
    message_row.exo_dt_status = helper.createRichText(text)
```

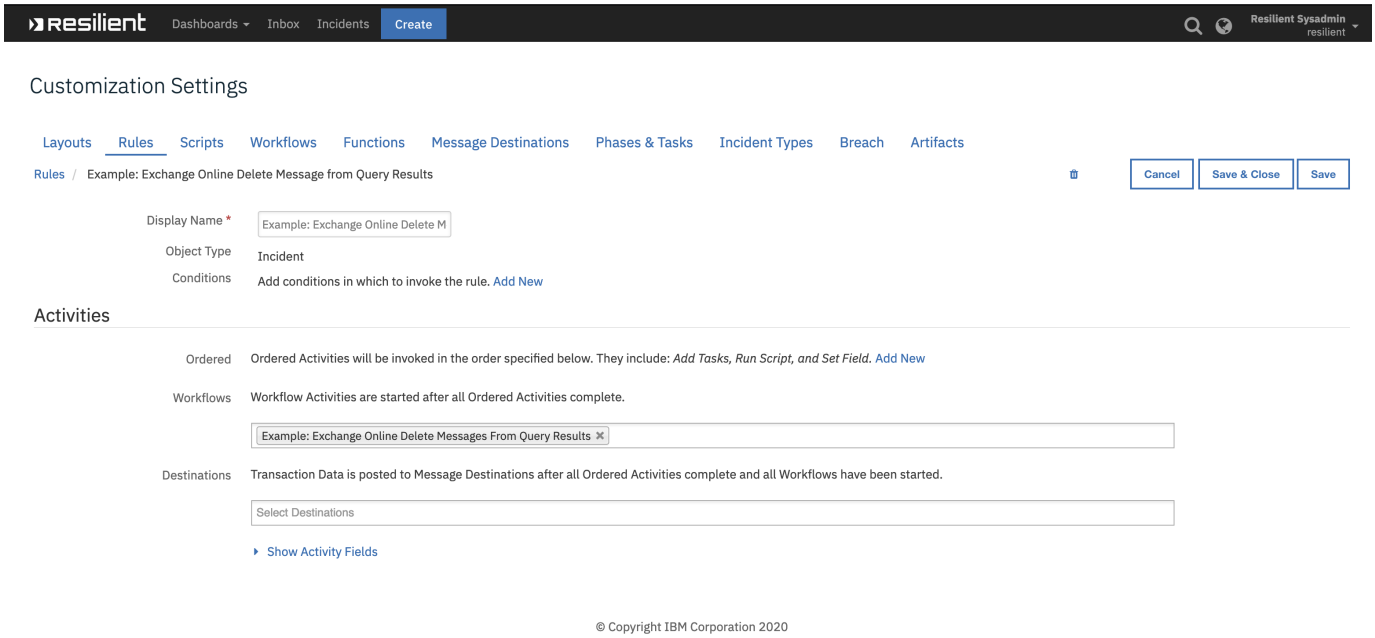
```
# Post a note containing the number of emails deleted
note = u"Exchange Online Delete Messages From Query Results:\n  {0} messages deleted".format(deleted_count)

# Add to the note if any messages from the query were not deleted.
if not_deleted_count > 0:
  note2 = u"  {0} messages NOT deleted".format(not_deleted_count)
  note = u"{0}\n{1}".format(note, note2)
incident.addNote(note)
```

▼ Example Workflow Output:

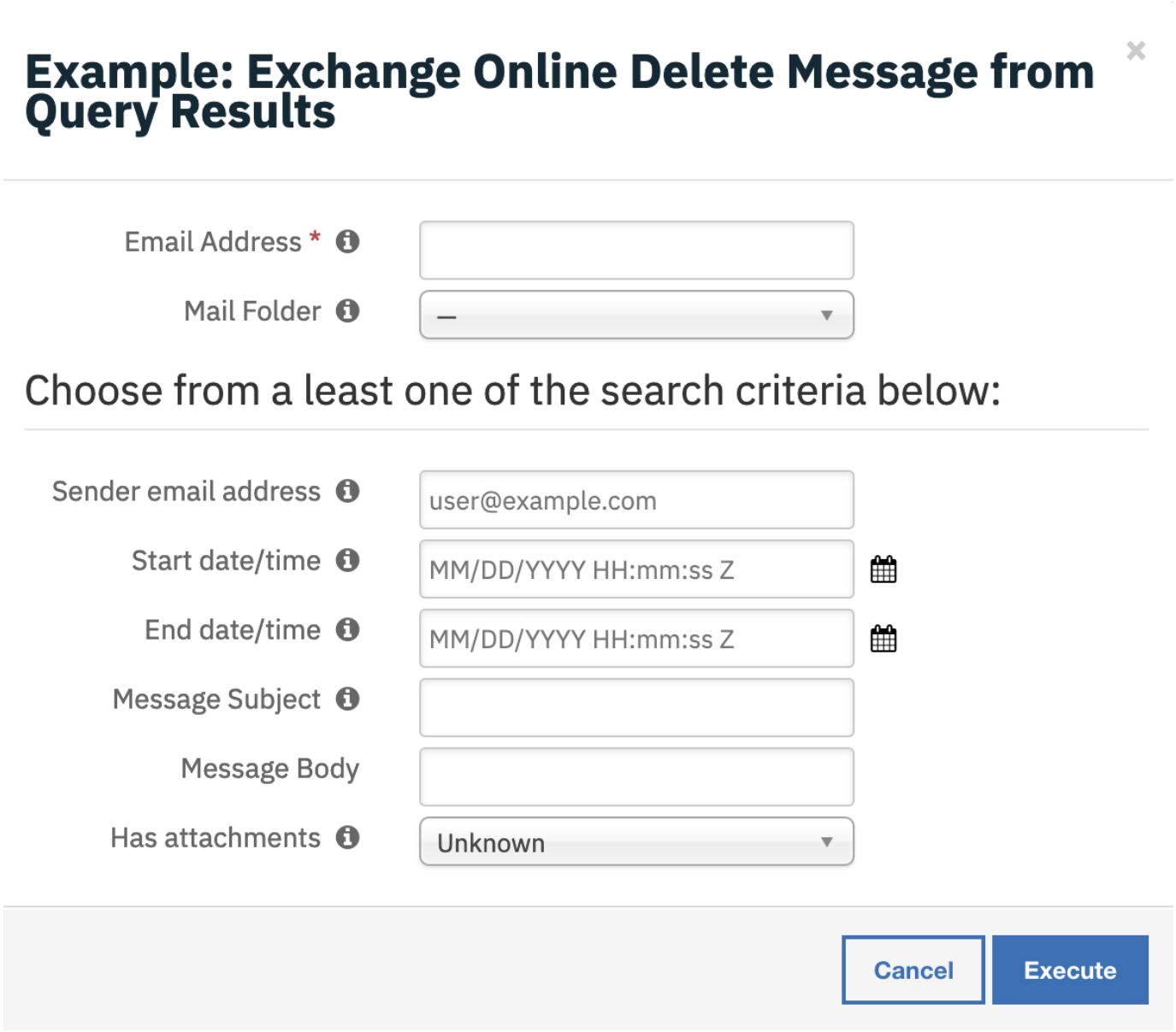
The messages deleted will appear in the data table with red "Deleted" text in the Status column:



▼ Example Rule:

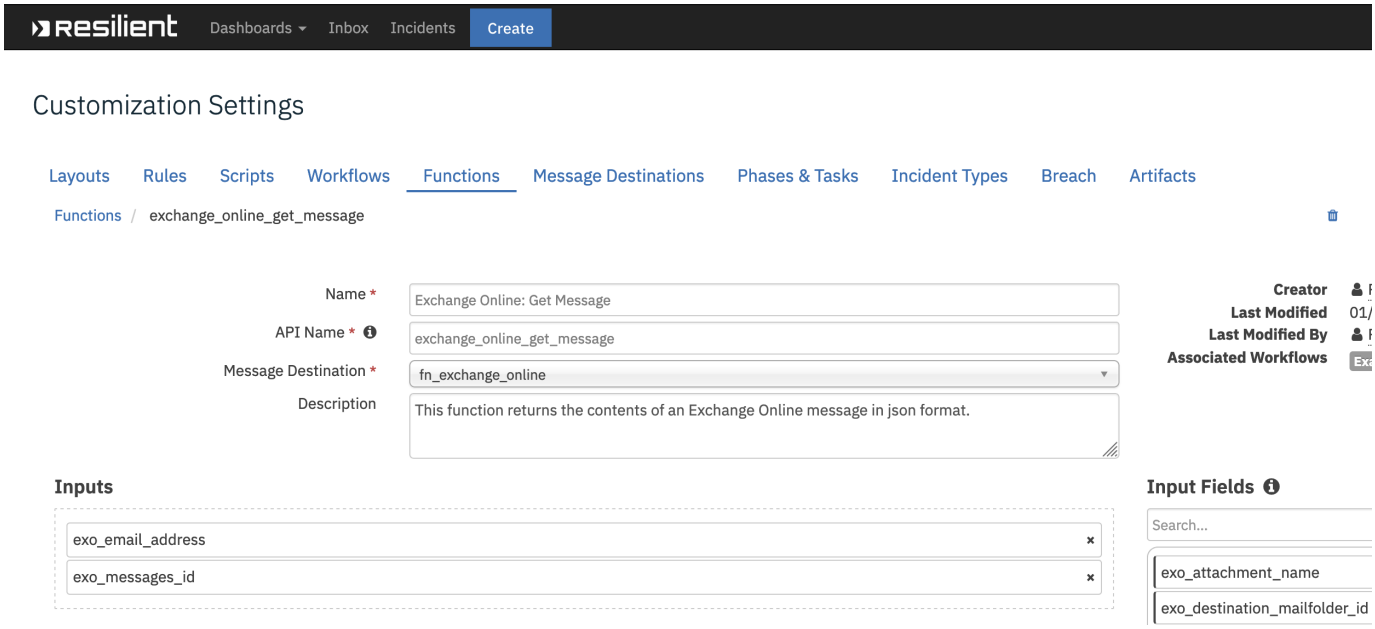


The Example: Exchange Online Delete Message From Query Results incident menu item rule will bring up the rule activity popup dialog pictured below to prompt for input querying message to be deleted. See the Query function section for a description of querying.



Function - Exchange Online: Get Message

This function returns the contents of an Exchange Online message in JSON format.



▼ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_messages_id	text	Yes	-	The message ID of the message to get

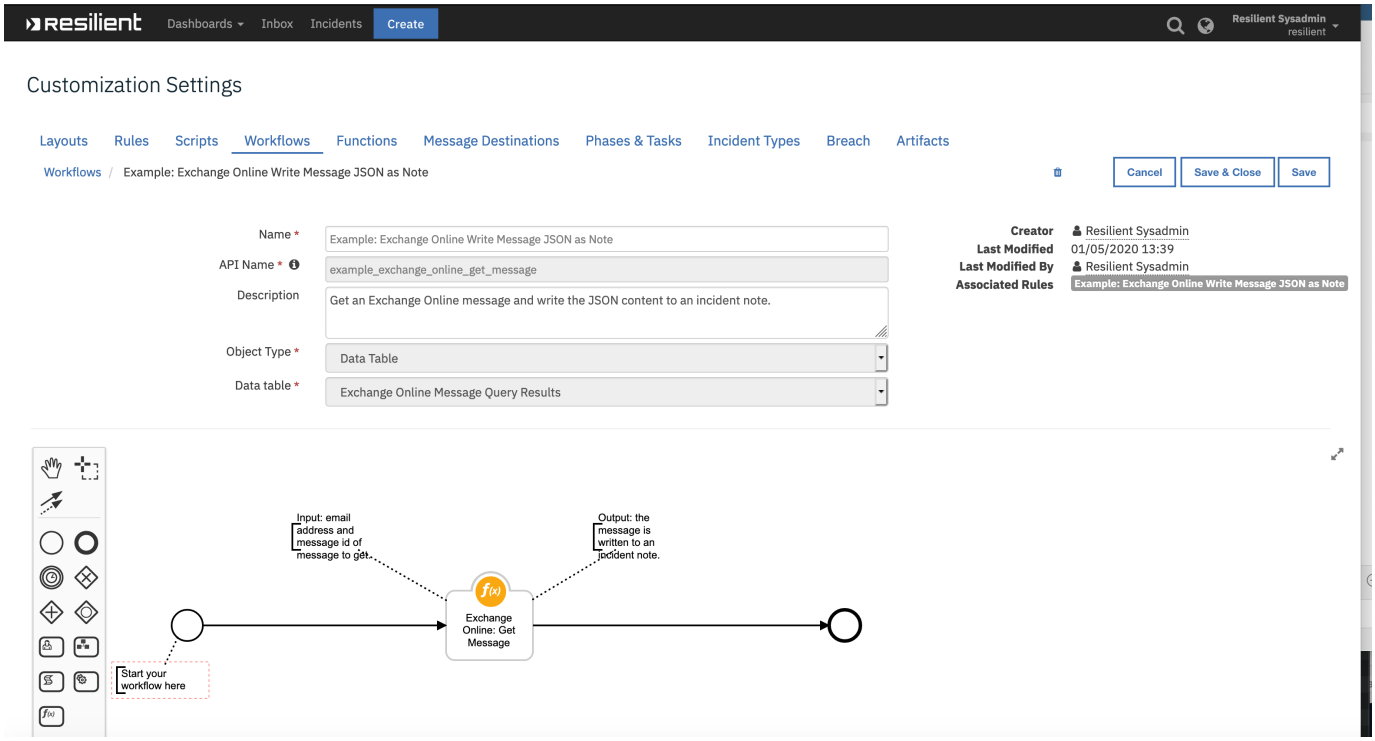
▼ Outputs:

```
results = {'inputs': {u'exo_messages_id': u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4
                      u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},
           'metrics': {'package': 'fn-exchange-online',
                       'timestamp': '2020-02-03 15:39:31',
                       'package_version': '1.0.0',
                       'host': 'cambridge.ibm.com',
                       'version': '1.0', '
                       'execution_time_ms': 654},
           'success': True,
           'pretty_string': u'{\n    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users(\'resilient2%4
           'content': {u'sentDateTime': u'2020-01-29T16:17:23Z',
                       u'conversationId':u'AAQkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAQAGjzfC_8ndROs7O00o-q
                       u'isDraft': False,
                       u'internetMessageId': u'<MMMMprod.outlook.com>',
                       u'id': u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwB
                       u'isReadReceiptRequested': False,
                       u'subject': u'test text area body',
                       u'lastModifiedDateTime': u'2020-02-03T16:46:41Z',
                       u'bodyPreview': u'test text area body',
                       u'from': {u'emailAddress': {u'name': u'Resilient User 3', u'address': u'resilient2@securitypocde
                       u'flag': {u'flagStatus': u'notFlagged'},
                       .
                       .
                       .
                       },
               'reason': None,
               'version': '1.0'}
```

▼ Workflows:

The workflow Write Message JSON as Note calls the Get Message function and writes the JSON contents returned from MS Graph API to an incident note. The JSON returned from MS Graph API contains information that is different from the .eml file, so both formats are provided in this package For information on writing EML to attachment see:

Write Message as Attachment.



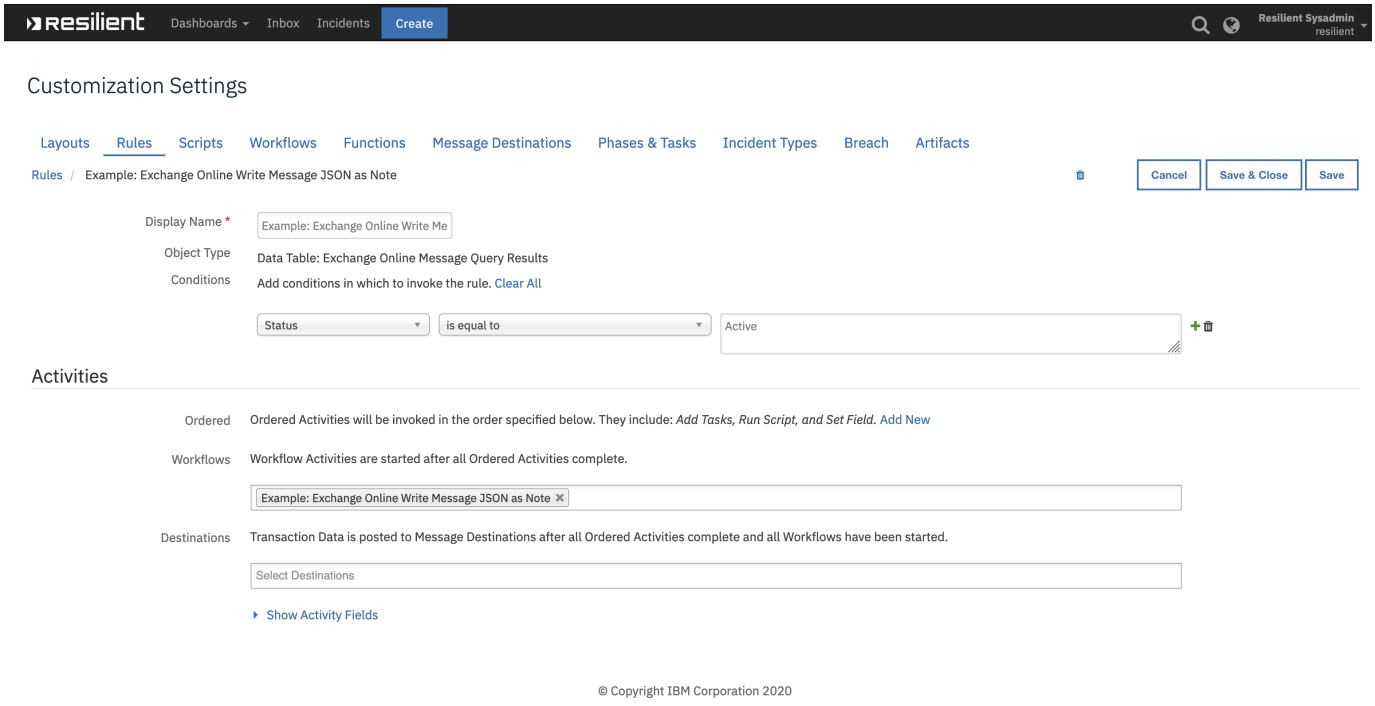
▼ Example Pre-Process Script:

```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
```

▼ Example Post-Process Script:

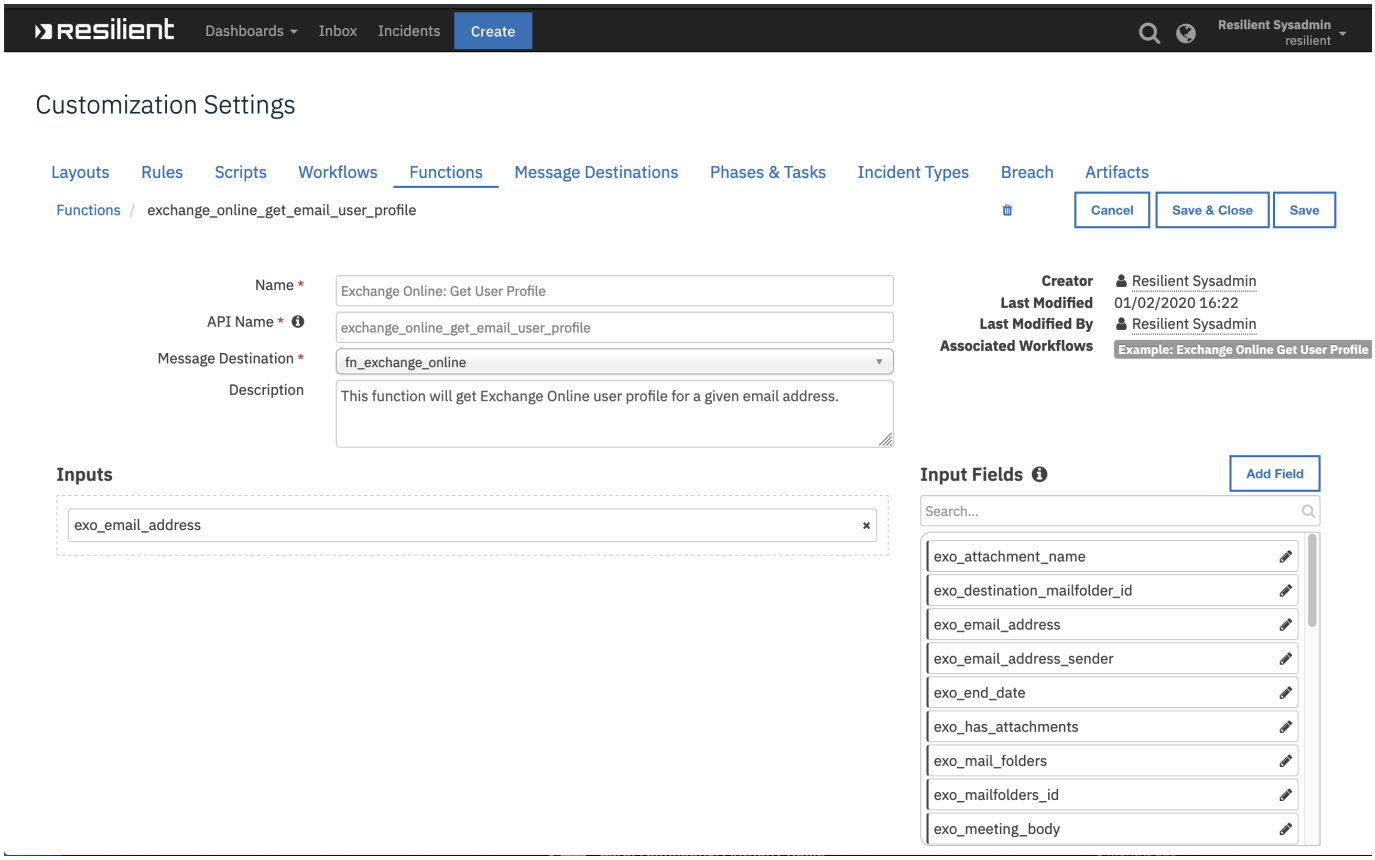
```
# Print the message to an incident note if it is found, otherwise update the status as Not Found in the datatable.
if results.content["error"] is not None:
  noteText = u"Exchange Online message NOT FOUND: \n email address: {0}\n message ID: {1}\n{2}".format(results.inputs[
  row.exo_dt_status = "Not Found"
else:
  noteText = u"Exchange Online email address: {0} message:\n{1}".format(results.inputs["exo_email_address"], results.p

incident.addNote(noteText)
```

▼ Example Rule:



Function - Exchange Online: Get User Profile

The Get User Profile function returns Exchange Online user profile for a given email address.



▼ Inputs:

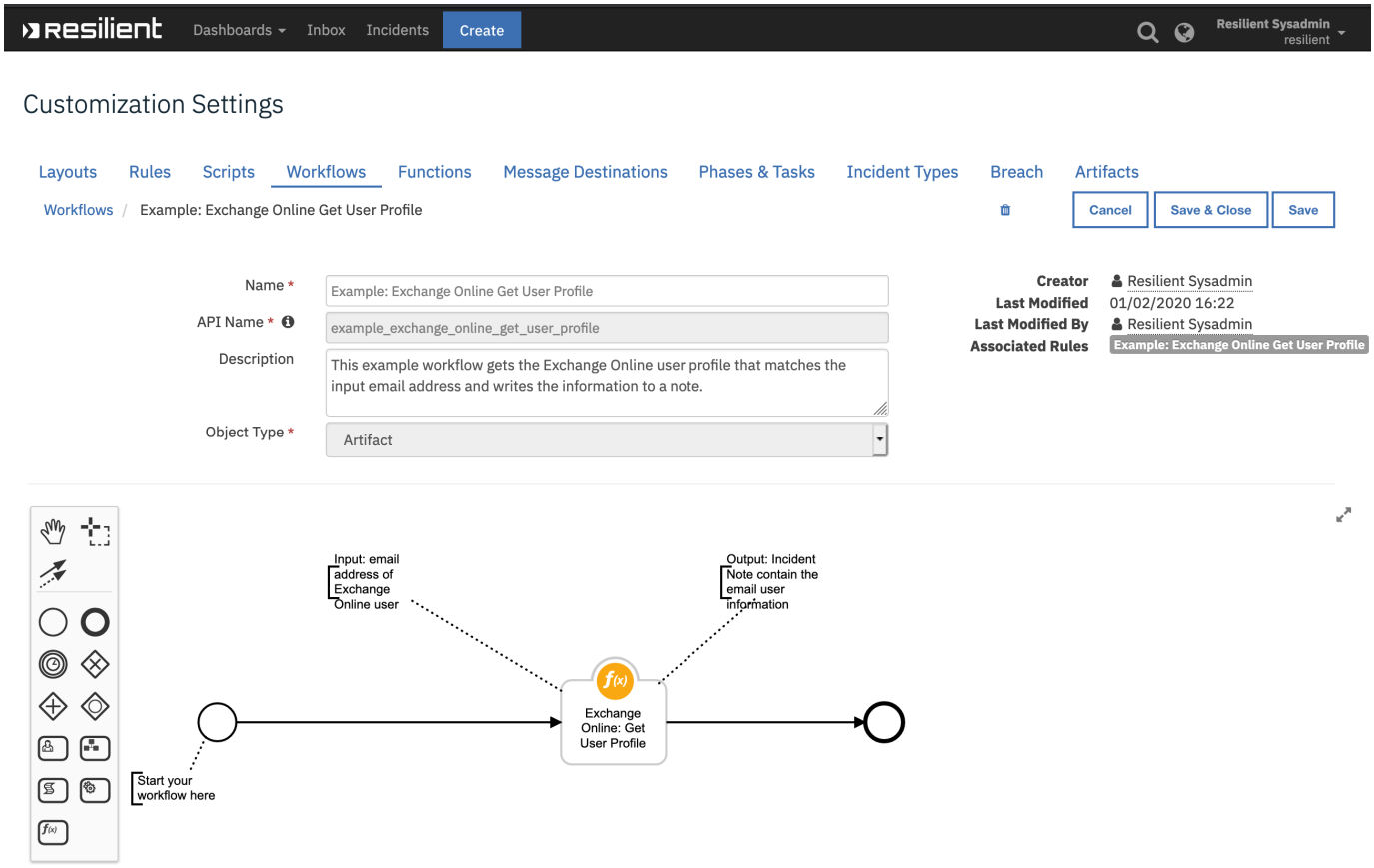
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account

▼ Outputs:


```
results = {
    'inputs': {u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},
    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-01-31 11:14:42',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 599},
    'success': True,
    'pretty_string': u'{\n    "@odata.context": "https://graph.microsoft.com/v1.0/$metadata#users/$entity",\n    "busi
    'content': {
        u'displayName': u'Resilient User 2',
        u'mobilePhone': None,
        u'preferredLanguage': u'en-US',
        u'jobTitle': None,
        u'userPrincipalName': u'resilient2@securitypocdemos.onmicrosoft.com',
        u'@odata.context': u'https://graph.microsoft.com/v1.0/$metadata#users/$entity',
        u'officeLocation': None,
        u'businessPhones': [],
        u'mail': u'resilient2@securitypocdemos.onmicrosoft.com',
        u'surname': u'Resilient User 2',
        u'givenName': u'Resilient User 2',
        u'id': u'393c1ebb-8222-4ba1-8665-f54eaf7f024f'},
    'raw': '{"displayName": "Resilient User 2", "mobilePhone": null, "preferredLanguage": "en-US", "jobTitle": null, "
    'reason': None,
    'version': '1.0'
}
```


▼ Workflows:

The example Get User Profile workflow accesses the artifact whose value contains the email address of the user whose profile is to be queried. The user profile is returned in JSON format as an incident note.



▼ Example Pre-Process Script:

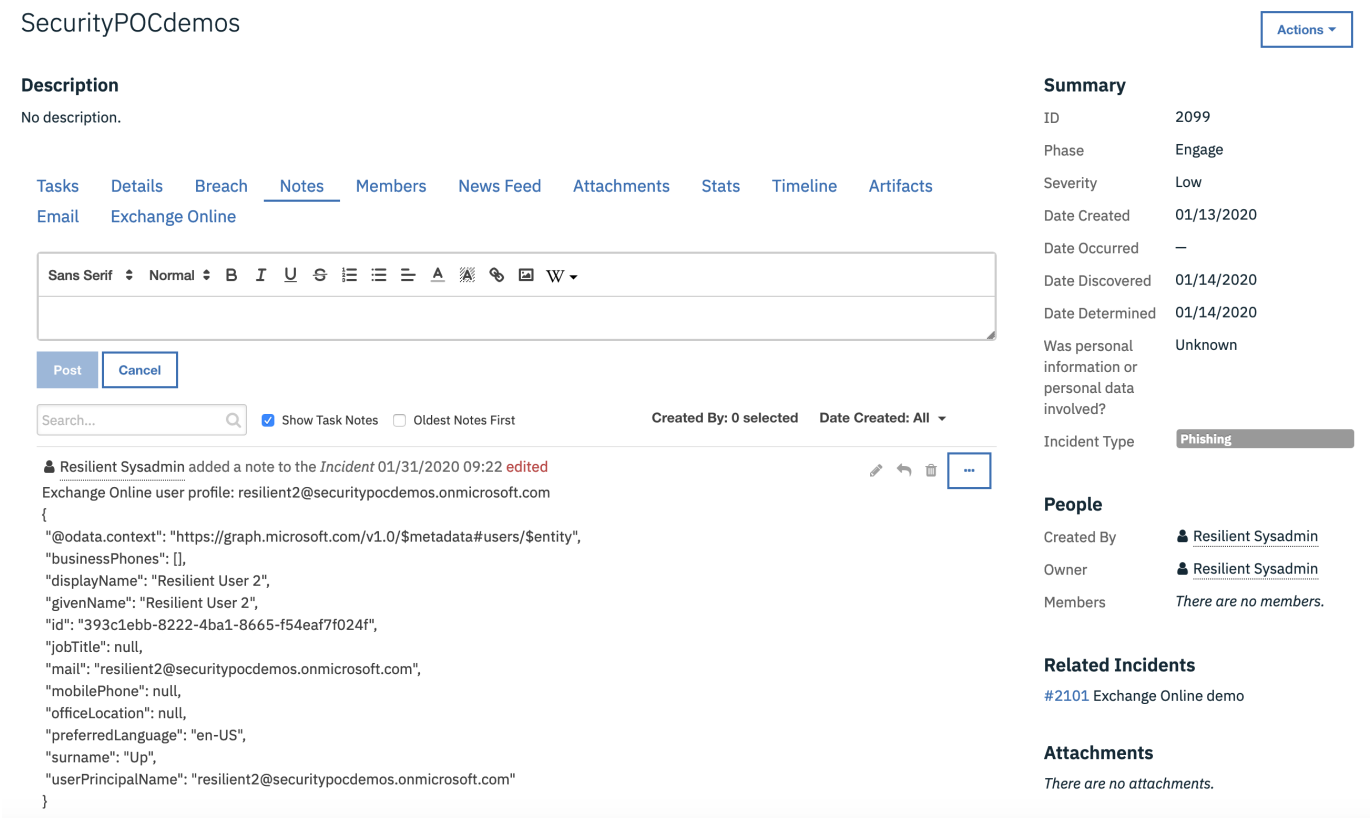
```
inputs.exo_email_address = artifact.value
```

▼ Example Post-Process Script:

```
if results.content["error"] is not None:
  noteText = u"Exchange Online user profile NOT FOUND: {0}\n{1}".format(results.inputs["exo_email_address"], results.p
else:
  noteText = u"Exchange Online user profile: {0}\n{1}".format(results.inputs["exo_email_address"], results.pretty_stri

incident.addNote(noteText)
```

▼ Example Workflow Output:

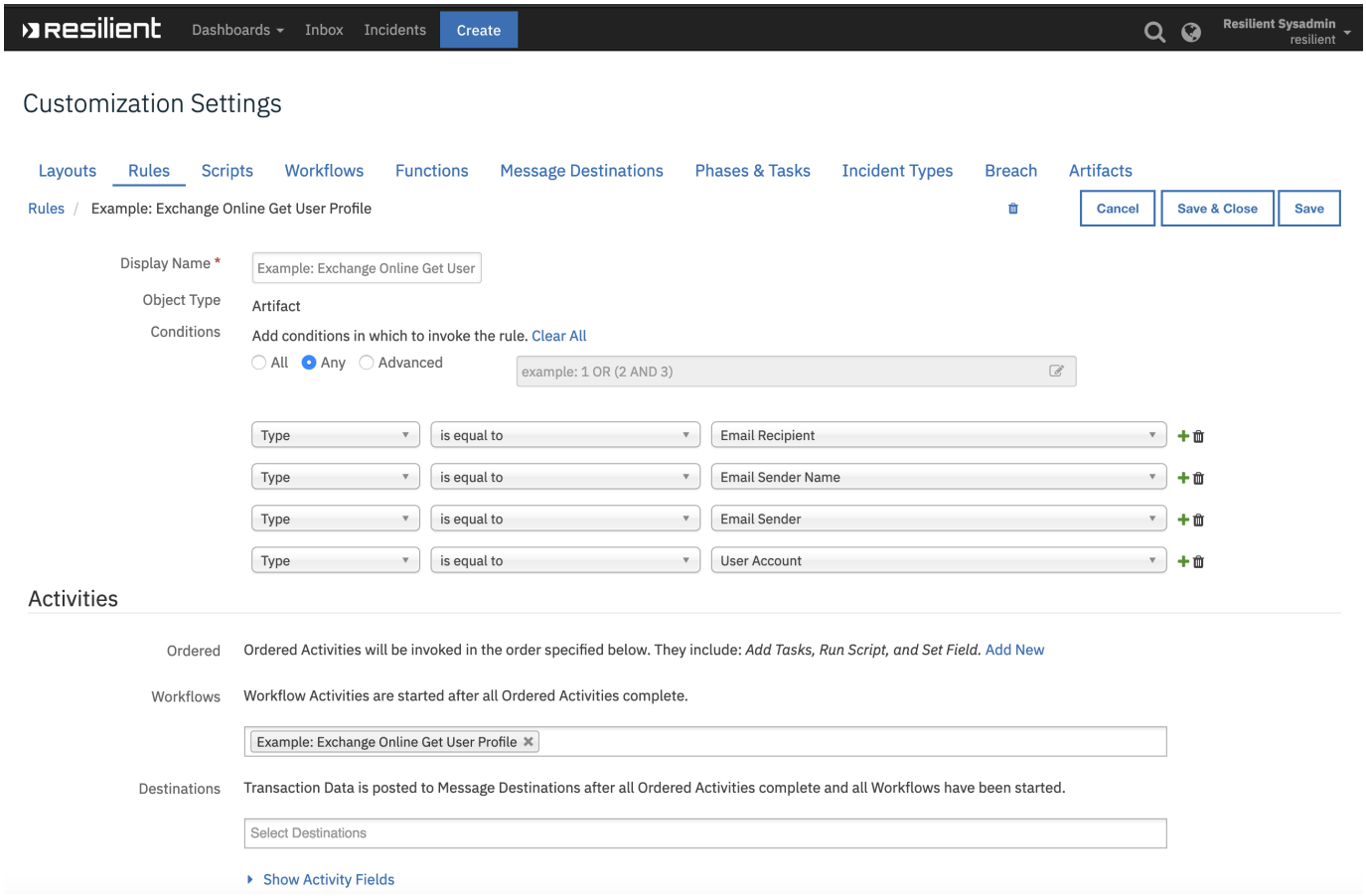
The Get User Profile workflow writes the user profile in JSON format to an incident note. Sample output of the note is pictured below:



▼ Example Rule:

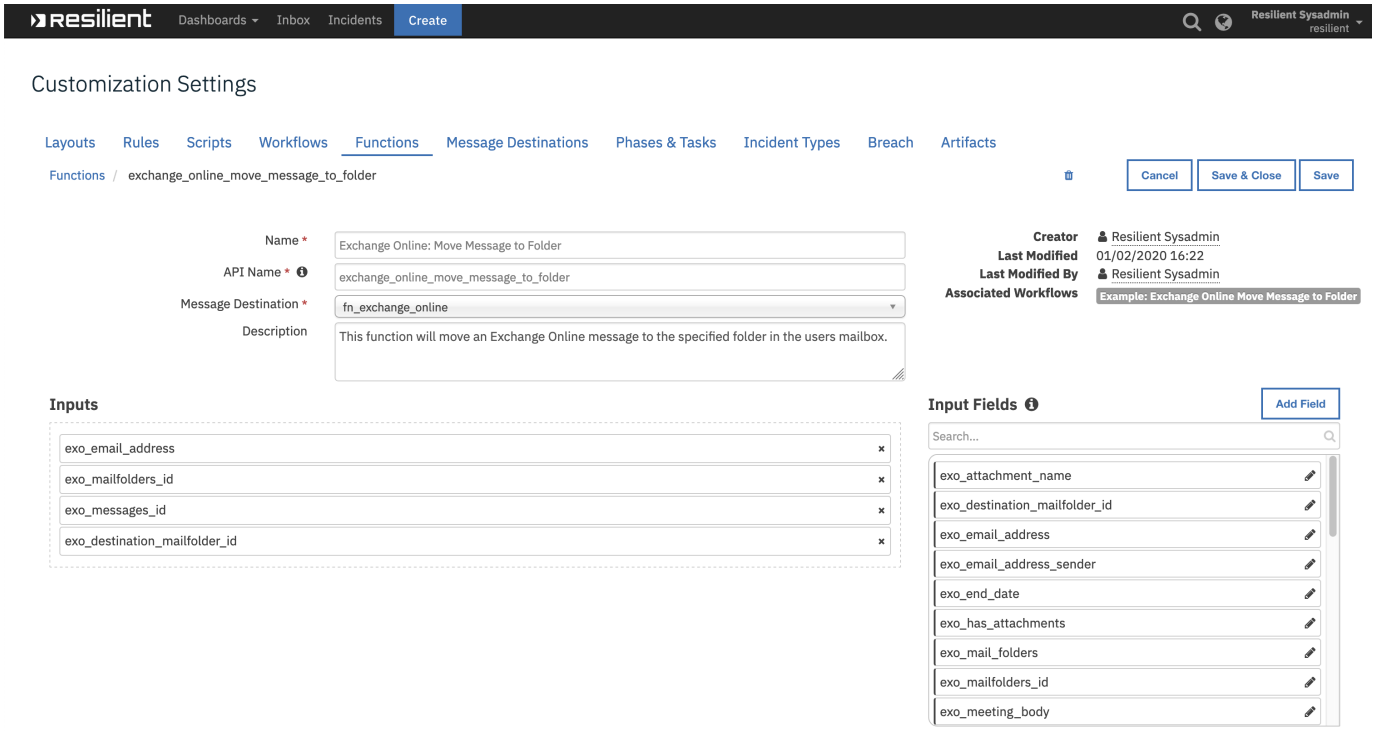
The example Get User Profile rule invokes the Get User Profile workflow if the artifact type is one of the following:

- Email Recipient
- Email Sender
- Email Sender Name
- User Account



Function - Exchange Online: Move Message to Folder

This function moves an Exchange Online message to the specified folder in the users mailbox.



▼ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_mailfolders_id	text	No	–	MailFolders ID
exo_messages_id	text	Yes	–	The message ID of the message to be deleted
exo_destination_mailfolder_id	select	Yes	recoverableitemsdeletions	Destination folder to which message is moved

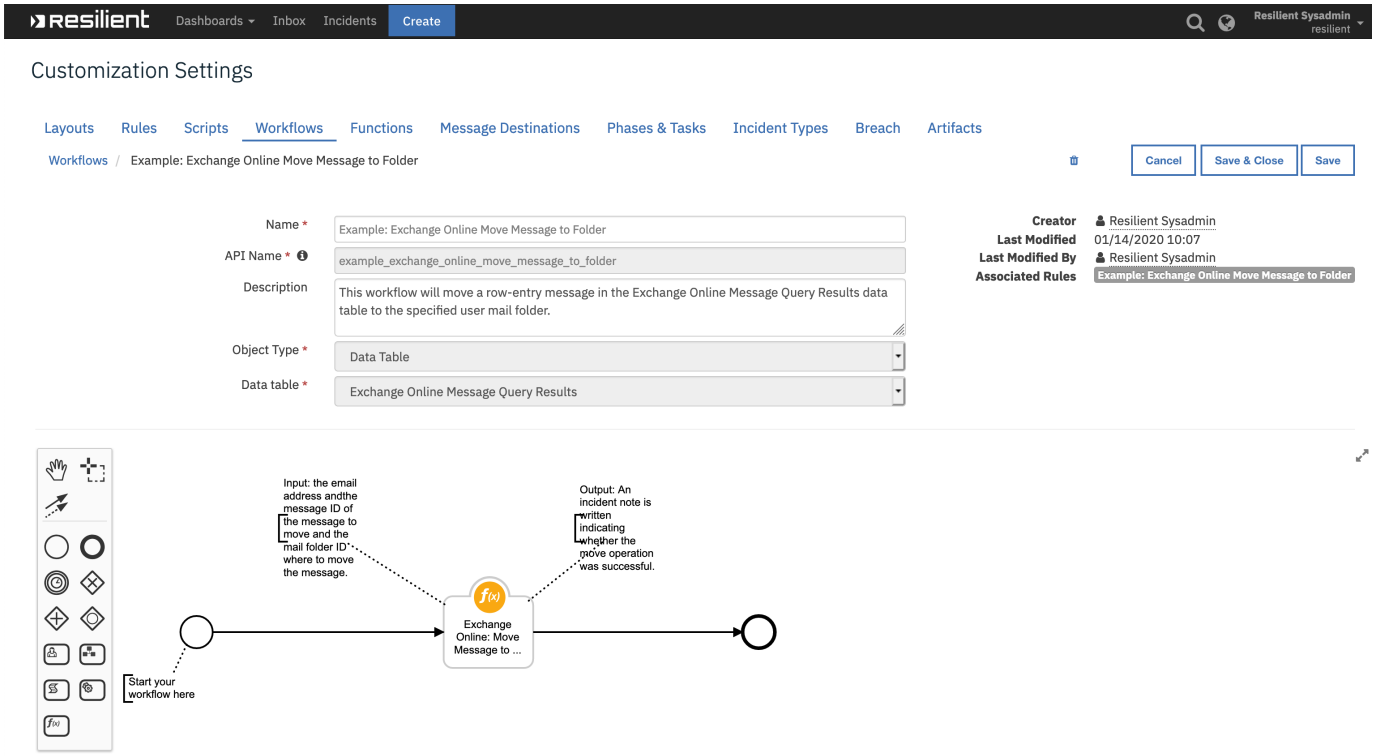
▼ Outputs:


```
Result: {
  'inputs': {u'exo_destination_mailfolder_id': {u'id': 126,
                                                u'name': u'recoverableitemsdeletions'},
             u'exo_mailfolders_id': None,
             u'exo_messages_id':u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSE
  'metrics': {'package': 'fn-exchange-online',
              'timestamp': '2020-01-31 13:23:41',
              'package_version': '1.0.0',
              'host': 'MacBook-Pro.local',
              'version': '1.0',
              'execution_time_ms': 1706},
  'success': True,
  'content': {'value': True},
  'raw': '{"value": true}',
  'reason': None,
  'version': '1.0'}
```


▼ Workflows:

The example Move Message to Folder workflow accesses the Exchange Online Message Query Results data table. Messages can be moved to a "Well known" Outlook folder, such as:

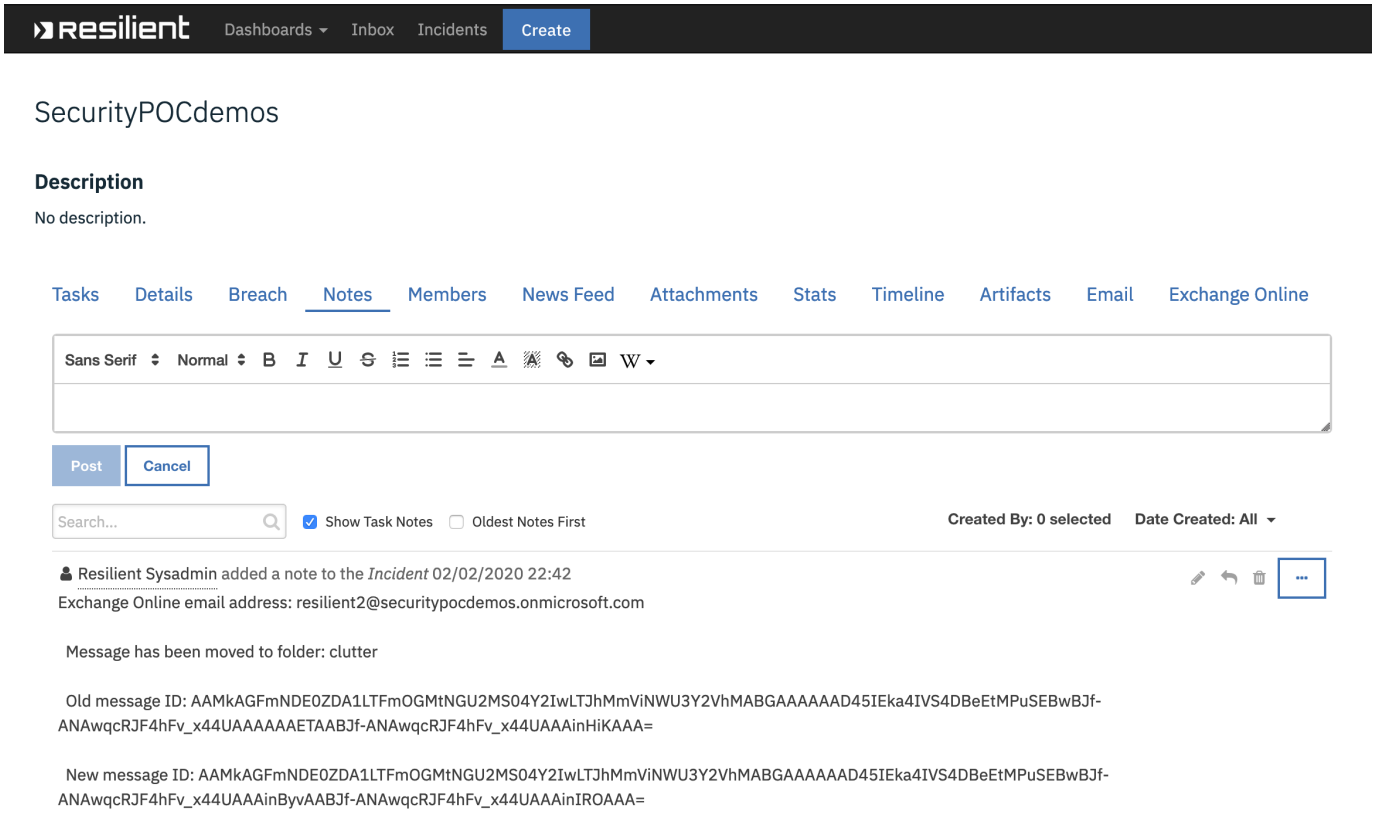
- archive
- clutter
- conflicts
- conversationhistory
- deleteditems
- drafts
- inbox
- junkemail
- localfailures
- msgfolderroot
- outbox
- recoverableitemsdeletions
- scheduled
- searchfolders
- sentitems



▼ Example Workflow Output:

The Move Message to Folder workflow calls the MS Graph API to move a message in the data table row to the Well-known Outlook folder specified in the exo_destination_mailfolder_id input parameter. When a message is moved, its message ID is changed, the status field column of the Exchange Online Message Query Results data table is updated to "Moved" in Red, and any rules to be run on the data table row are changed.

Below is a screen shot of an example Note after a message is moved to a folder:



▼ Example Pre-Process Script:

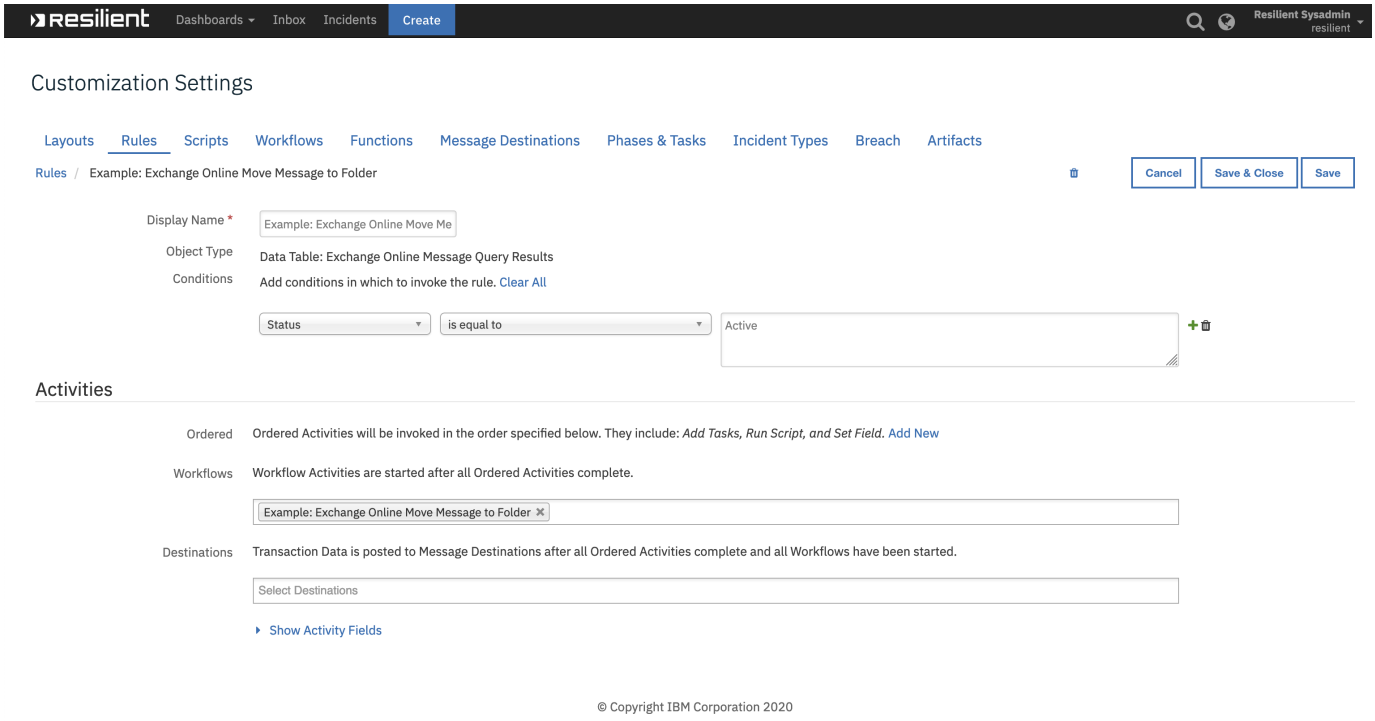
```
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_mailfolders_id = None
inputs.exo_messages_id = row.exo_dt_message_id
inputs.exo_destination_mailfolder_id = rule.properties.exo_destination_mailfolder_id
```

▼ Example Post-Process Script:

```
if results.content["error"] is not None:
  # Print the message to an incident note if it is found, otherwise update the status as Not Found in the datatable.
  noteText = u"Exchange Online message NOT FOUND: \n email address: {0}\n message ID: {1}".format(results.inputs["exo_
  status_text = u"""<p style= "color:{color}">{status} </p>""".format(color="red", status="Not Found")
  row['exo_dt_status'] = helper.createRichText(status_text)
else:
  # When a message is moved it's ID changes, so update the new message ID into the data table
  noteText = u"Exchange Online email address: {0}\n\n  Message has been moved to folder: {1}\n\n  Old message ID: {2}
  row['exo_dt_message_id'] = results.content["new_message_id"]
incident.addNote(noteText)
```

▼ Example Rule:

The example Move Message to Folder rule accesses the Exchange Online Message Query Results data table. When the status column of the row is Active the Move Message To Folder rule is available to initiate the corresponding workflow. The status column may be non-Active if the message is Deleted or Not Found, in which case the message cannot be moved.



Function - Exchange Online: Query Messages

The Exchange Online: Query Message function queries the Exchange Online to find messages matching the specified input parameters. The function returns a list of messages matching the search criteria.

The function will search over the following email accounts:

- all mailboxes of a tenant (specify "all", "ALL", "all users")
- single email address
- comma separated list of email addressed

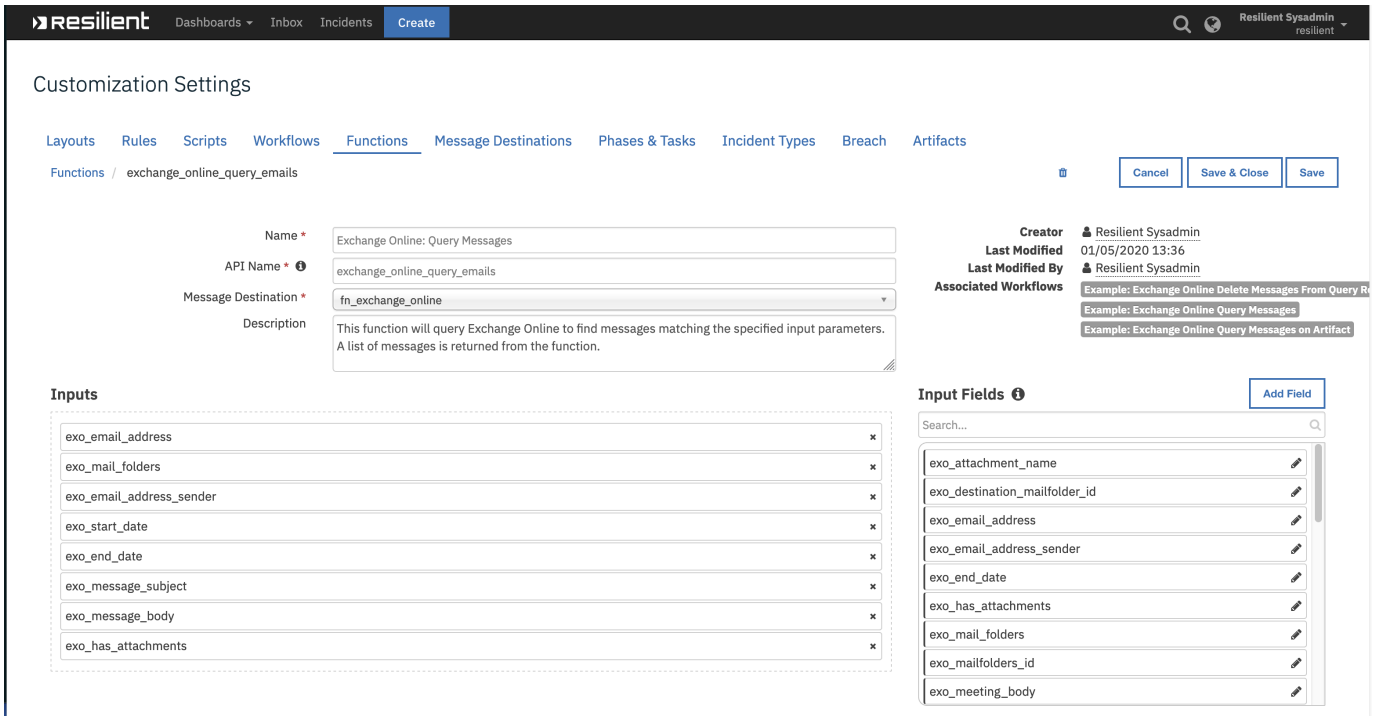
If no mail folder if specified, all folders and subdirectories are queried. The mail folder to be searched can be one of the list of Outlook "Well-known" folders:

- archive
- clutter
- conflicts
- conversationhistory
- deleteditems
- drafts
- inbox
- junkemail
- localfailures
- msgfolderroot
- outbox
- recoverableitemsdeletions
- scheduled
- searchfolders
- sentitems

At least one of the following search criteria must be passed to the query messages function:

- Sender email address
- Message received date/time start/end
- Message subject "contains" text
- Message body "contains" text
- Boolean flag indicating whether the message has an attachment

NOTE: There can be a large number of results of the Query Message function. If needed, use the max_user and max_messages parameters in the app.config file to limit the number of users searched and the number of messages returned from a query.



▼ Inputs:

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_email_address_sender	text	No	user@example.com	Search messages sent from this email address; leave blank to ignore sender attribute
exo_end_date	datetimepicker	No	-	Query messages received ending at this date/time
exo_has_attachments	boolean	No	-	True to include attachments, False to exclude attachments, Unknown to get all
exo_mail_folders	text	No	Inbox	The folder to search in the users mailbox
exo_message_body	text	No	message body text	message body
exo_message_subject	text	No	message subject	message subject

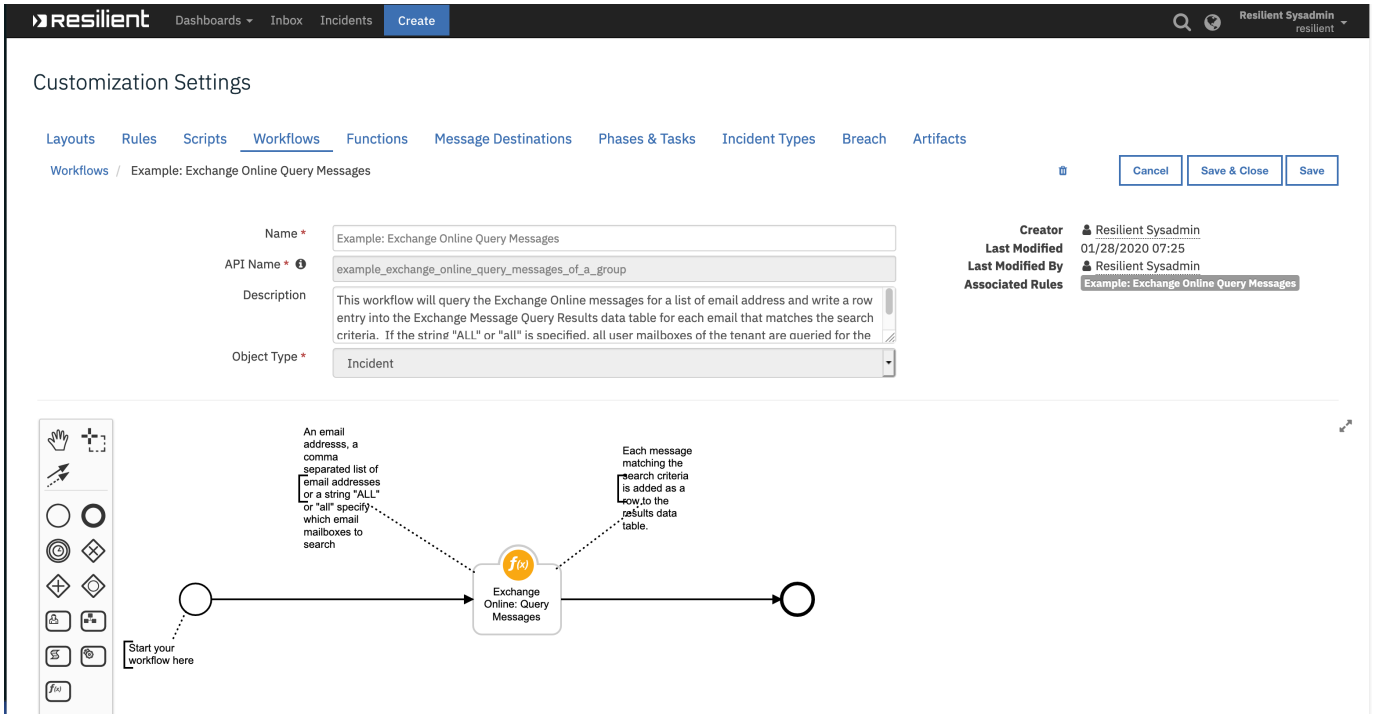
Name	Type	Required	Example	Tooltip
exo_start_date	datetimepicker	No	–	Query messages received starting at this date/time

▼ Outputs:


```
results = {
   'inputs': {u'exo_start_date': None,
              u'exo_email_address_sender': None,
              u'exo_end_date': None,
              u'exo_message_subject': u'lunch',
              u'exo_has_attachments': None,
              u'exo_email_address': u'all',
              u'exo_mail_folders': None,
              u'exo_message_body': None},
    'metrics': {'package': 'fn-exchange-online',
              'timestamp': '2020-02-04 15:36:12',
              'package_version': '1.0.0',
              'host': 'MacBook-Pro.local',
              'version': '1.0',
              'execution_time_ms': 9492},
    'success': True,
    'content': [{'status_code': 200,
                 'email_address': u'resilient2@securitypocdemos.onmicrosoft.com',
                 'email_list': [{u'sentDateTime': u'2020-02-04T20:44:02Z',
                 u'conversationId': u'AAQkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAQANIxI-VmHPNIslAVFC4yWrI='
                 u'internetMessageId': u'<MWHPR2201MB11359DA0C07AF09AB319DD9FB1030@MWHPR2201MB1135.namprd22.prod.outloo
                 u'id': u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4DBeEtMPuSEBwBJf-ANA
                 u'subject': u'lunch',
                 u'lastModifiedDateTime': u'2020-02-04T20:44:04Z',
                 u'bodyPreview': u"Let's have lunch at 12!",
                 u'from': {u'emailAddress': {u'name': u'Resilient User 2',
                           u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
                 u'flag': {u'flagStatus': u'notFlagged'},
                 u'isDraft': False,
                 u'replyTo': [],
                 u'changeKey': u'CQAAABYAAABJf/ANAwqcRJF4hFv+x44UAAAlac/G', u'receivedDateTime': u'2020-02-04T20:44:03Z
                 u'parentFolderId': u'AQMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMAAuAAAD_OSBJGuCFUuAwXhLTD7kh
                 u'body': {u'content': u'<html>\r\n<head>\r\n<meta http-equiv="Content-Type" content="text/html; charse
                 u'isDeliveryReceiptRequested': False,
                 u'importance': u'normal',
                 u'toRecipients': [{u'emailAddress': {u'name': u'Resilient User 3',
                                   u'address': u'resilient3@securitypocdemos.onmicrosoft.com'}}],
                 u'ccRecipients': [],
                 u'isRead': True,
                 u'categories': [],
                 u'sender': {u'emailAddress': {u'name': u'Resilient User 2',
                             u'address': u'resilient2@securitypocdemos.onmicrosoft.com'}},
                 u'createdDateTime': u'2020-02-04T20:44:02Z',
                 u'webLink': u'https://outlook.office365.com/owa/?ItemID=AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNW
                 u'conversationIndex': u'AQHV25vV0jEj9WYc80iyUBUULjJasg==',
                 u'hasAttachments': False,
                 u'bccRecipients': [],
                 u'inferenceClassification': u'focused',
                 u'@odata.etag': u'W/"CQAAABYAAABJf/ANAwqcRJF4hFv+x44UAAAlac/G"'}]}],
      'reason': None,
      'version': '1.0'
}
```


▼ Workflows:

The Example: Exchange Online Query Messages workflow places results into the Exchange Online Query Results data table, which appears on the Exchange Online custom incident tab. (See the Installation Guide to configure the custom tab.)



▼ Example Pre-Process Script:

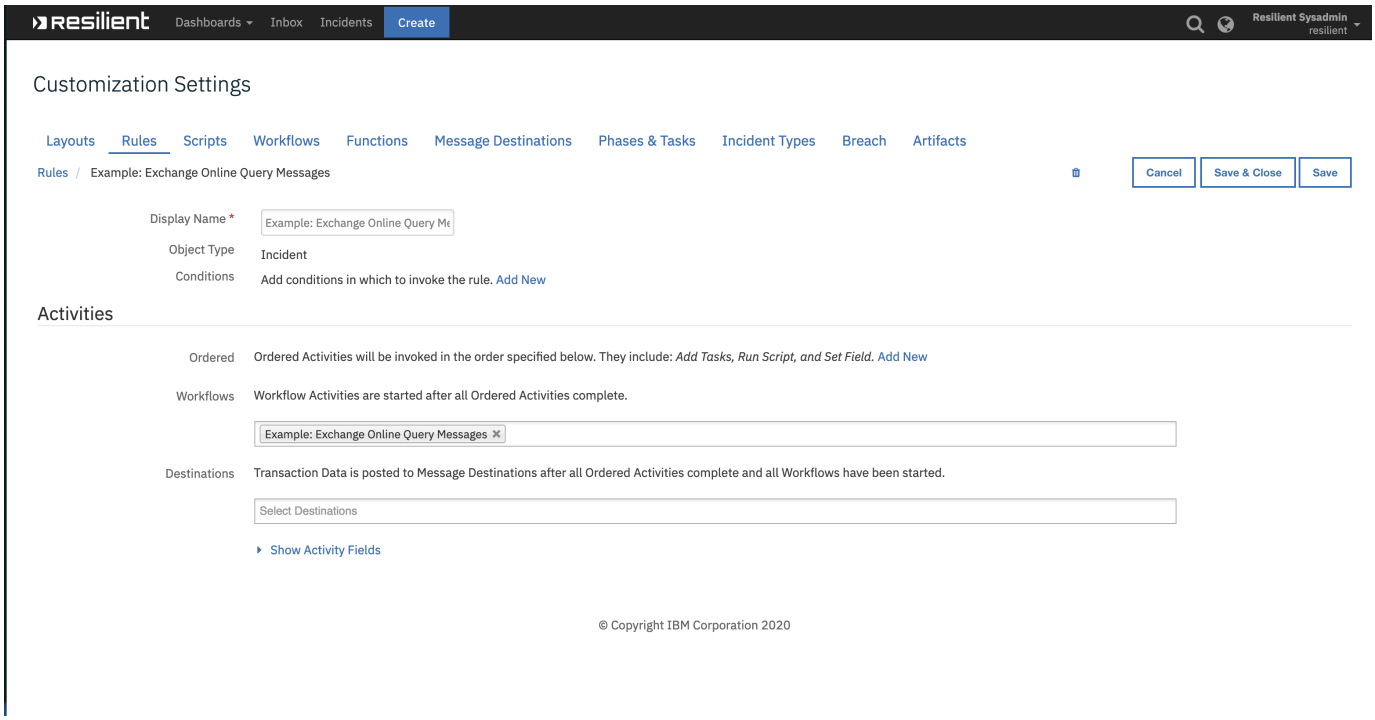
```
# Get the email address of the user whose mailbox will be queried.
inputs.exo_email_address = inputs.exo_email_address if rule.properties.exo_email_address_list is None else rule.proper

# Get the search criteria from the activity rules if available.
inputs.exo_mail_folders         = inputs.exo_mail_folders         if rule.properties.exo_mailfolder_id        is None
inputs.exo_email_address_sender = inputs.exo_email_address_sender if rule.properties.exo_email_address_sender is None
inputs.exo_message_subject      = inputs.exo_message_subject      if rule.properties.exo_message_subject      is None
inputs.exo_message_body         = inputs.exo_message_body         if rule.properties.exo_message_body         is None
inputs.exo_start_date           = inputs.exo_start_date           if rule.properties.exo_start_date           is None
inputs.exo_end_date             = inputs.exo_end_date             if rule.properties.exo_end_date             is None
inputs.exo_has_attachments      = inputs.exo_has_attachments      if rule.properties.exo_has_attachments      is None
```

▼ Example Post-Process Script:

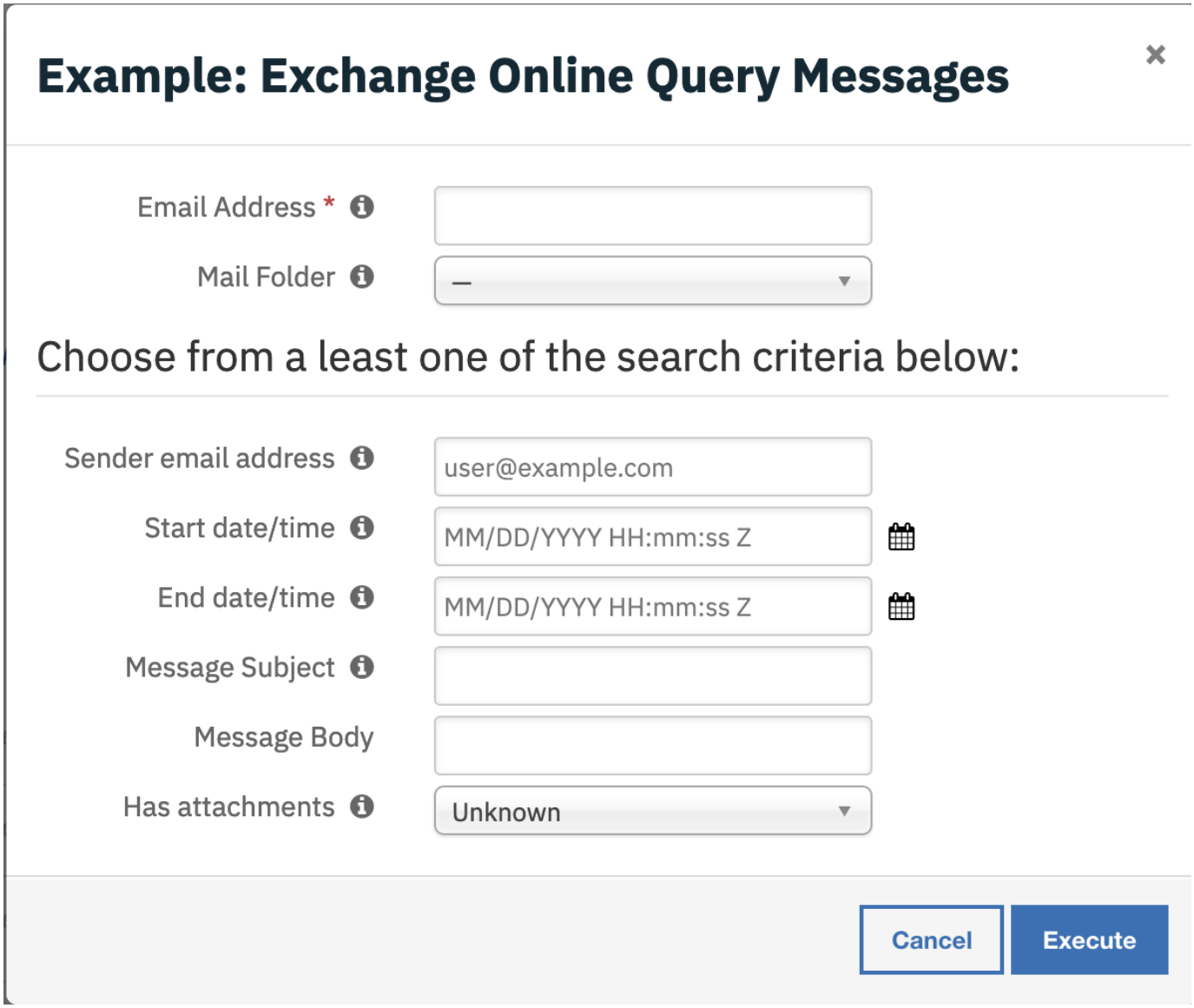
```
from java.util import Date

note = u"Exchange Online Query Multiple users:\n"
note_len = len(note)

# Add each email as a row in the query results data table
for user in results["content"]:
  # If an email address is not found post to a note.
  if user["status_code"] == 404:
    line = u"email address not found: {}\n".format(user["email_address"])
    note = note + line

  for email in user["email_list"]:
    message_row = incident.addRow("exo_message_query_results_dt")
    message_row.exo_dt_query_date = Date()
    message_row.exo_dt_message_id = email.id
    message_row.exo_dt_received_date   = email.receivedDateTime
    message_row.exo_dt_email_address = user["email_address"]
    if email.sender:
      message_row.exo_dt_sender_email = email.sender.emailAddress.address
    else:
```

```
      message_row.exo_dt_sender_email = ""
    message_row.exo_dt_message_subject = email.subject
    message_row.exo_dt_has_attachments = email.hasAttachments
    if email.webLink:
      ref_html = u"""<a href='{0}'>Link</a>""".format(email.webLink)
      message_row.exo_dt_web_link = helper.createRichText(ref_html)
    else:
      message_row.exo_dt_web_link = ""

    message_row.exo_dt_status = helper.createRichText("Active")

# If any email addresses where not found post a note
if len(note) > note_len:
  incident.addNote(note)
```

▼ Example Rule:

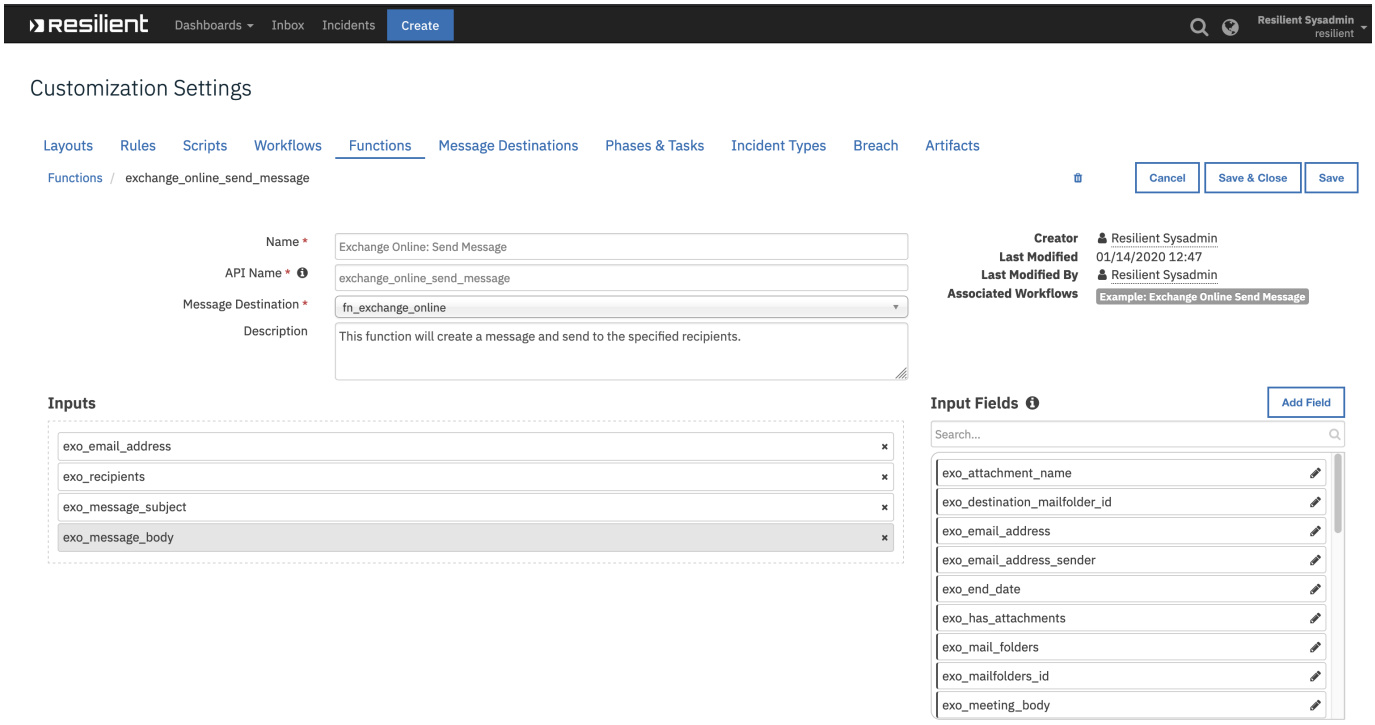


When the Example: Exchange Online Delete Messages from Query Results rule is activated, the following rule activity popup dialog prompts for input to create the meeting and send a message to the invitees:



Function - Exchange Online: Send Message

This function creates a message and sends it to the specified recipients.



▼ Inputs:

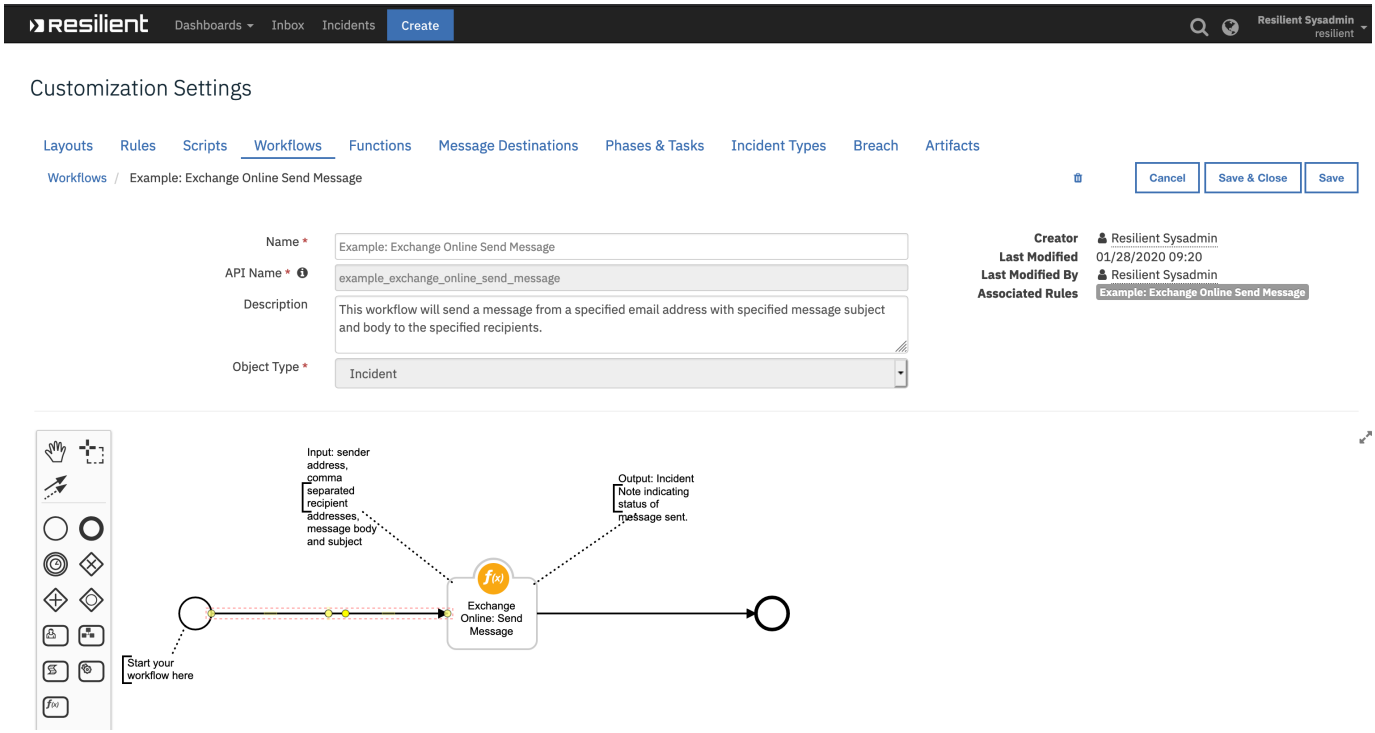
Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_message_body	text	No	message body text	Message body
exo_message_subject	text	No	message subject	Message subject
exo_recipients	text	Yes	–	Comma separated list of message recipients

▼ Outputs:

```
results = {
    'inputs': {u'exo_recipients': u'resilient2@securitypocdemos.onmicrosoft.com,        resilient3@securitypocdemos.on
               u'exo_message_subject': u'Please investigate',
               u'exo_message_body': u'<div class="rte"><div>Can you look into this?</div><div><br /></div><div>Thanks!<
               u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},

    'metrics': {'package': 'fn-exchange-online',
                'timestamp': '2020-02-04 11:18:16',
                'package_version': '1.0.0',
                'host': 'MacBook-Pro.local',
                'version': '1.0',
                'execution_time_ms': 796},
    'success': True,
    'content': {'value': True},
    'raw': '{"value": true}',
    'reason': None,
    'version': '1.0'}
```

▼ Workflows:

The Example: Exchange Online Send Message workflow calls the Exchange Online Send Message function and writes an incident note containing the results of the function.



▼ Example Pre-Process Script:

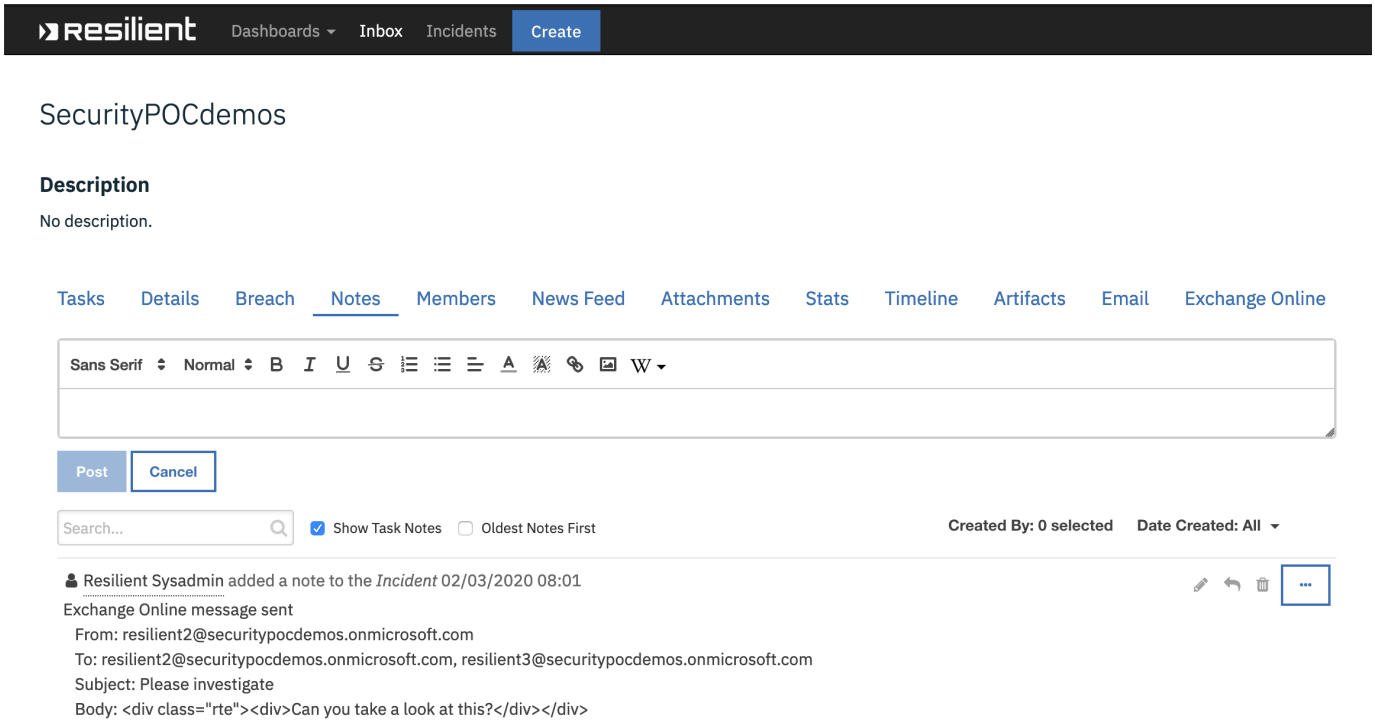
```
inputs.exo_email_address   = inputs.exo_email_address   if rule.properties.exo_message_sender_address is None else rul
inputs.exo_recipients      = inputs.exo_recipients      if rule.properties.exo_message_recipients     is None else rul
inputs.exo_message_subject = inputs.exo_message_subject if rule.properties.exo_message_subject is None else rule.prope
inputs.exo_message_body    = inputs.exo_message_body    if rule.properties.exo_message_send_body.content is None else
```

▼ Example Post-Process Script:

```
if results.success:
  noteText = u"Exchange Online message sent\n   From: {0}\n   To: {1}\n   Subject: {2}\n   Body: {3}".format(results.i
else:
  noteText = u"Exchange Online message NOT sent\n   From: {0}\n  To: {1}".format(results.inputs["exo_email_address"],

incident.addNote(noteText)
```

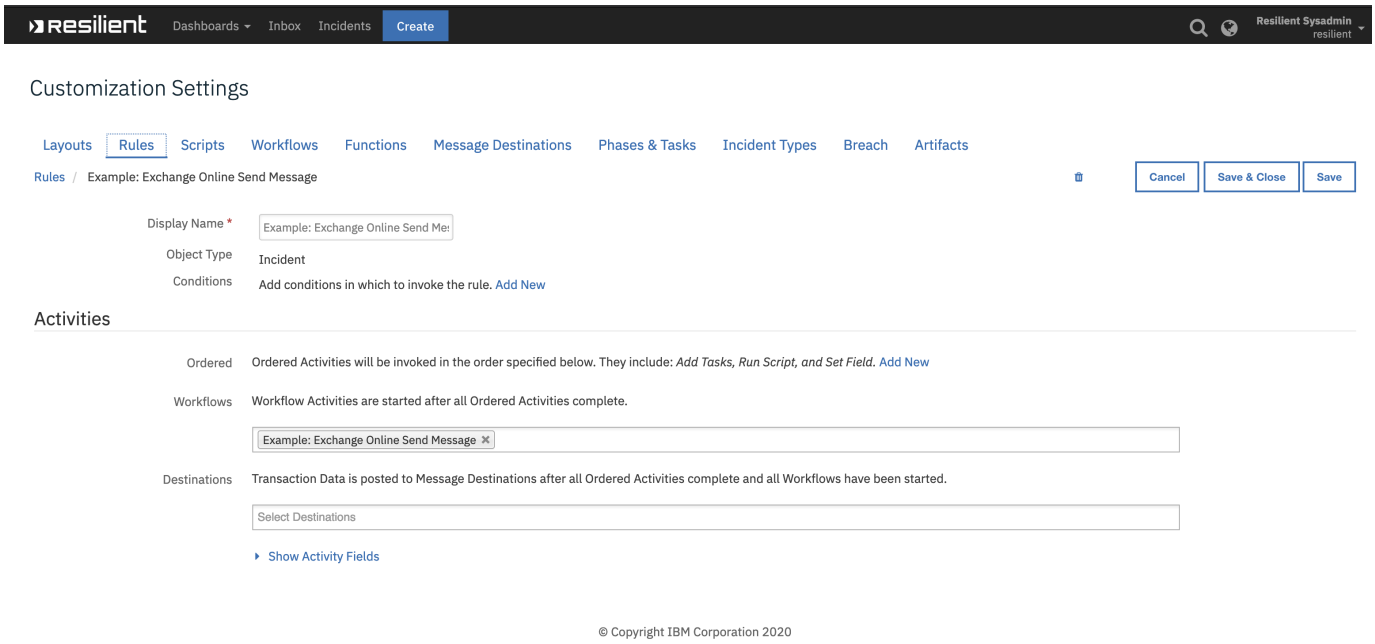
▼ Example Workflow Output:

The following is a sample incident note that is created from Example: Exchange Online Send Message workflow:

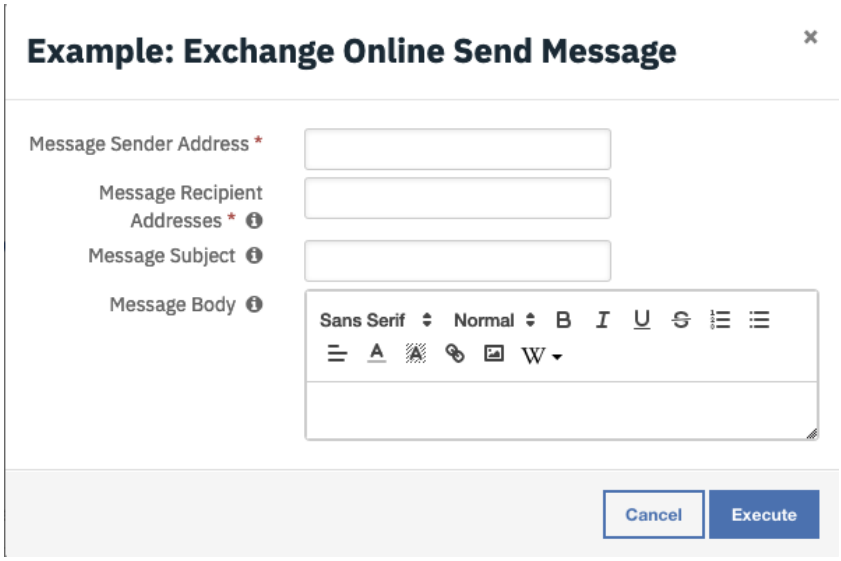


▼ Example Rule:

The following Example: Exchange Online Send Message incident menu item rule is included to send a message via Exchange Online:

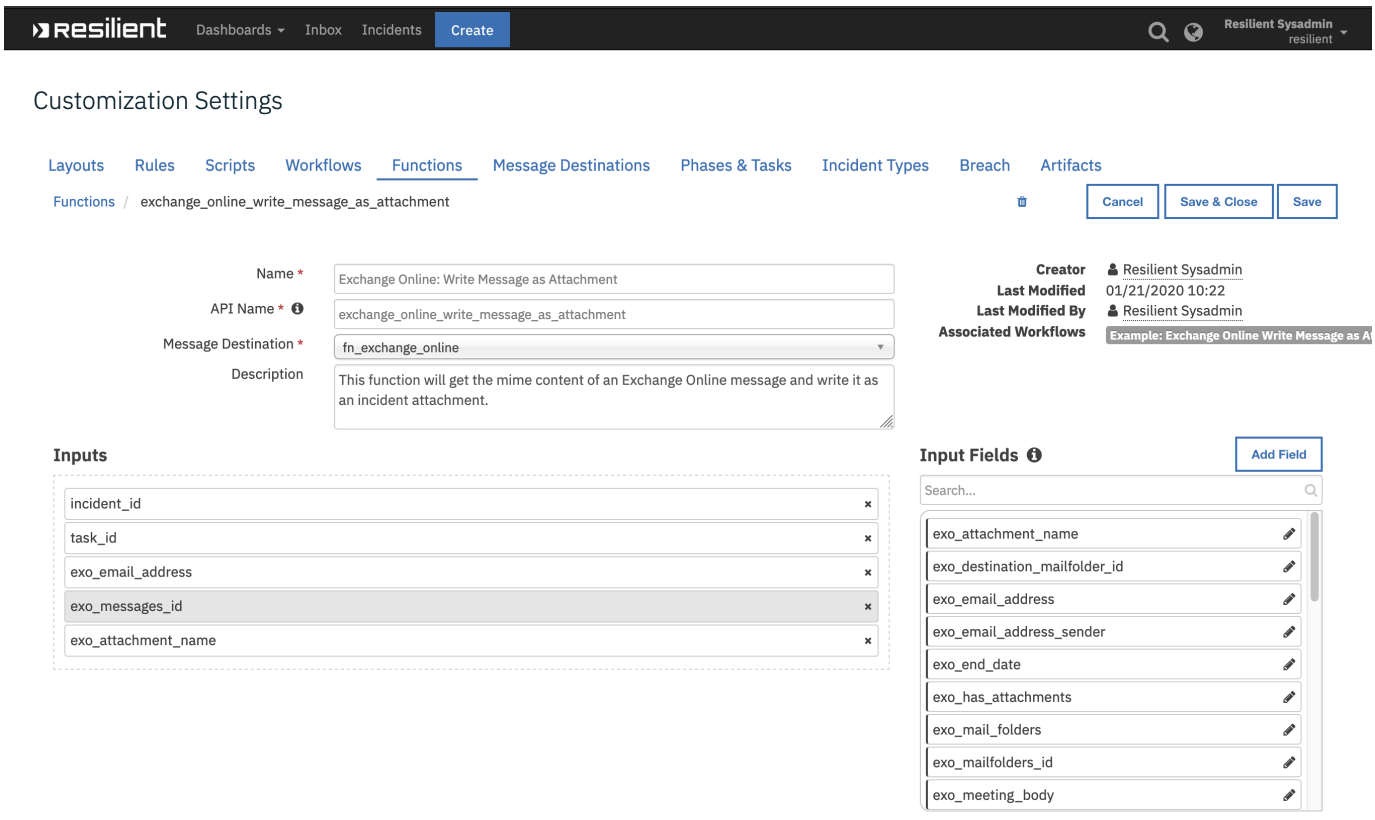


When the Example Send Message rule is initiated the following rule activity popup dialog will appear prompting for input on the message to send:



Function - Exchange Online: Write Message as Attachment

This function gets the mime content of an Exchange Online message and writes it as an incident attachment. The attachment file name is an optional parameter. The function uses a default message-{email-address}-{message-ID}.eml filename if none is specified.



▼ Inputs:

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

Name	Type	Required	Example	Tooltip
exo_email_address	text	Yes	user@example.com	Get information on this user email account
exo_messages_id	text	Yes	-	The message ID of the message to be deleted
exo_attachment_name	text	No	my-message.eml	The attachment file to which message is written

▼ Outputs:

```
results = {'inputs': {u'incident_id': 2099,
                      u'exo_attachment_name': u'my-message.eml',
                      u'exo_messages_id': u'AAMkAGFmNDE0ZDA1LTFmOGMtNGU2MS04Y2IwLTJhMmViNWU3Y2VhMABGAAAAAAD45IEka4IVS4
                       u'exo_email_address': u'resilient2@securitypocdemos.onmicrosoft.com'},
           'metrics': {'package': 'fn-exchange-online',
                       'timestamp': '2020-02-03 14:54:13',
                       'package_version': '1.0.0',
                       'host': 'annmarie-mbp.cambridge.ibm.com',
                       'version': '1.0', 'execution_time_ms': 5929},
            'success': True, 'content': {'attachment_name': u'my-message.eml'},
            'raw': '{"attachment_name": "my-message.eml"}',
            'reason': None,
            'version': '1.0'}
}
```

▼ Workflows:

▼ Example Pre-Process Script:

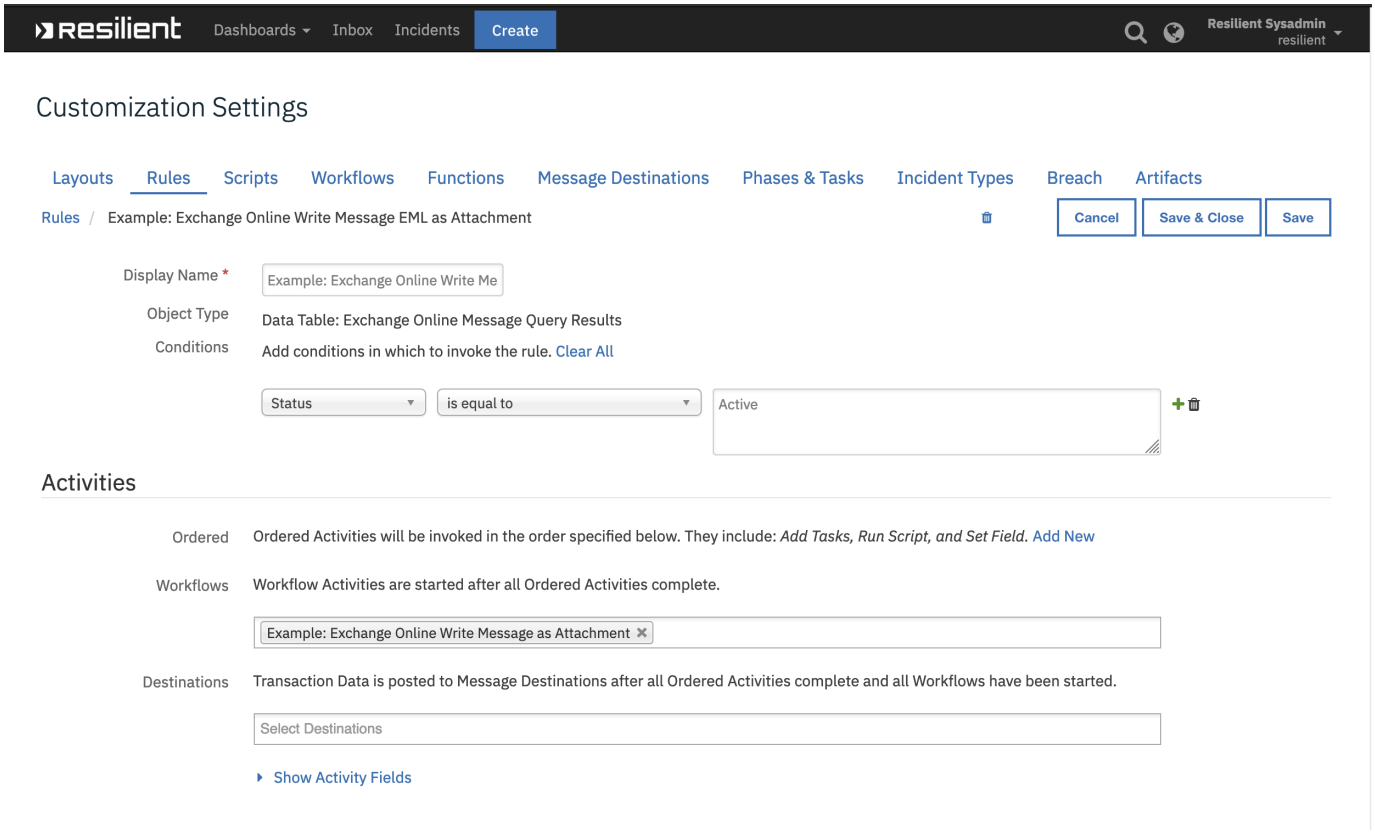
```
inputs.incident_id = incident.id
#inputs.task_id = task.id
inputs.exo_attachment_name = rule.properties.exo_attachment_name
inputs.exo_email_address = row.exo_dt_email_address
inputs.exo_messages_id = row.exo_dt_message_id
```

▼ Example Post-Process Script:

```
None
```

▼ Example Rule:

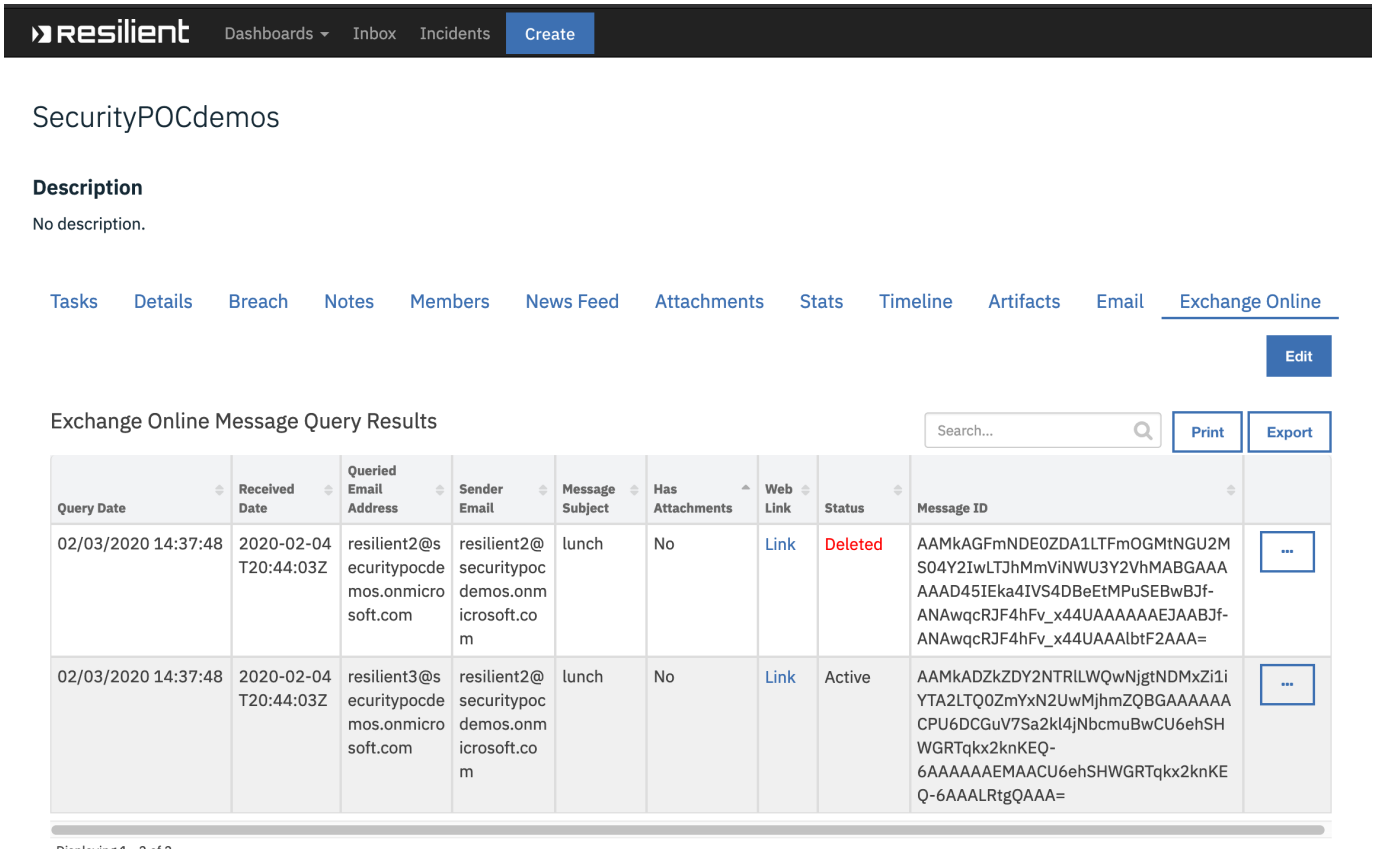
The example Write Message EML as Attachment rule accesses the Exchange Online Message Query Results data table. When the status column of the row is Active, the Write Message as Attachment rule is available to initiate the corresponding workflow. If the status is non-Active, which includes Deleted or Not Found, the message content cannot be retrieved or written.



Data Table - Exchange Online Message Query Results

The following is an example of message query results that are populated in the Exchange Online Message Query Results data table:

NOTE: The Web Link column contains a link to the message, but you must be logged in as the message owner user to view the message in the link.



API Name:

exo_message_query_results_dt

Columns:

Column Name	API Access Name	Type	Tooltip
Queried Email Address	`exo_dt_email_address`	`text`	-
Has Attachments	`exo_dt_has_attachments`	`boolean`	-
Message ID	`exo_dt_message_id`	`text`	-
Message Subject	`exo_dt_message_subject`	`text`	-
Query Date	`exo_dt_query_date`	`datetimepicker`	-
Received Date	`exo_dt_received_date`	`text`	-
Sender Email	`exo_dt_sender_email`	`text`	-
Status	`exo_dt_status`	`textarea`	-
Web Link	`exo_dt_web_link`	`textarea`	-

Rules

Rule Name	Object	Workflow Triggered
Example: Exchange	exo_message_query_results_dt	`example_exchange_online_get_message`

Rule Name	Object	Workflow Triggered
Online Write Message JSON as Note		
Example: Exchange Online Send Message	incident	`example_exchange_online_send_message`
Example: Exchange Online Get User Profile	artifact	`example_exchange_online_get_user_profile`
Example: Exchange Online Delete Message from Query Results	incident	`example_exchange_online_delete_messages_from_query_results`
Example: Exchange Online Move Message to Folder	exo_message_query_results_dt	`example_exchange_online_move_message_to_folder`
Example: Exchange Online Query Messages on Artifact	artifact	`example_exchange_online_query_emails`
Example: Exchange Online Write Message EML as Attachment	exo_message_query_results_dt	`example_exchange_online_write_message_as_attachment`
Example: Exchange Online Query Messages	incident	`example_exchange_online_query_messages_of_a_group`
Example: Exchange Online Create Meeting	incident	`example_exchange_online_create_meeting`
Example: Exchange Online Create Artifacts	exo_message_query_results_dt	–
Example: Exchange Online Delete Message	exo_message_query_results_dt	`example_exchange_online_delete_email`