



A MITEL
PRODUCT
GUIDE

Mitel Workflow Studio

Mitel Workflow Studio Administrator Guide

Version 1.0.1

July 2025

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by **Mitel Networks Corporation (MITEL®)**. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos and graphics (collectively "Trademarks") appearing on Mitel's Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC), its affiliates, parents, or subsidiaries (collectively "Mitel") or others. Use of the Trademarks is prohibited without the express consent from Mitel. Please contact our legal department at legal@mitel.com for additional information. For a list of the worldwide Mitel Networks Corporation registered trademarks, please refer to the website: <http://www.mitel.com/trademarks>.

®,™ Trademark of Mitel Networks Corporation

© Copyright 2025, Mitel Networks Corporation

All rights reserved

Contents

1 Preface.....	1
1.1 Purpose.....	1
1.2 Intended Audience.....	1
1.3 Acronyms and Abbreviations.....	1
1.4 Terminologies.....	2
1.5 Licensing.....	3
1.6 Related Documents.....	3
 2 Mitel Workflow Studio – An Overview.....	 4
2.1 Sample Use Cases.....	4
2.2 Limitations.....	4
 3 Licensing.....	 5
3.1 Workflow Integrations.....	5
3.2 Flows, Actions, and Triggers.....	5
3.3 Workflow Dashboard Usage Reporting.....	6
3.4 License Usage Notifications and Overages.....	6
3.5 Customer Responsibilities.....	6
3.6 Subscription Expiry.....	7
 4 Getting Started.....	 8
4.1 Integrating Workflow Studio.....	8
4.2 Migrating Voice Assist to Workflow Studio.....	9
4.3 Logging In.....	10
4.4 Reporting a problem.....	11
4.5 Logging Out.....	12
4.6 Application Environment.....	12
 5 Understanding Workflow Studio.....	 15
5.1 Dashboard.....	15
5.1.1 What's New.....	16
5.1.2 Training.....	16
5.1.3 Top 5 Run Flows in current billing period.....	16
5.1.4 Licenses.....	17
5.2 Flows.....	18
5.2.1 Create a flow.....	19
5.2.2 Edit a flow.....	24
5.2.3 Import a flow.....	25
5.2.4 Clone a flow.....	25
5.2.5 Delete a single or multiple flows.....	26

5.3 Library.....	27
5.3.1 Actions.....	27
5.3.2 Triggers.....	34
5.3.3 Templates.....	42
5.3.4 Integrations.....	43

6 Workflow Management..... 55

6.1 Views.....	55
6.1.1 Editor.....	55
6.1.2 Running.....	57
6.1.3 Debug.....	60
6.2 Basic Operations.....	65
6.2.1 Building Workflows.....	66
6.2.2 Viewing Workflow.....	68
6.2.3 Import a Workflow.....	68
6.2.4 Export a Workflow.....	68
6.2.5 Resizing Workflow.....	69
6.2.6 Expanding/Collapsing Workflow.....	69
6.3 Activity Wheel.....	70
6.3.1 Media Activities.....	70
6.3.2 User Management Activities.....	72
6.3.3 Developer Activities.....	73
6.3.4 Routing Activities.....	75
6.3.5 MCX Activities.....	76
6.3.6 Messaging Activities.....	77
6.3.7 Integration Activities.....	79
6.3.8 Paste Activity.....	87
6.4 Expressions.....	87
6.4.1 Values.....	88
6.4.2 Variables.....	88
6.4.3 Arithmetic.....	89
6.4.4 Boolean.....	90
6.4.5 Equalities.....	90
6.4.6 Arrays.....	90
6.4.7 Objects.....	92
6.4.8 String Expressions.....	94
6.4.9 Date and Time Expressions.....	97
6.5 Using Variables.....	98
6.5.1 Add a Variable.....	99
6.5.2 Delete a Variable.....	99
6.5.3 Assigning Values to Variables.....	99

7 List of Actions..... 101

7.1 Media Actions.....	102
7.1.1 Dial menu.....	103
7.1.2 Greeting.....	106
7.1.3 Dial by name.....	108
7.1.4 Transfer.....	110
7.1.5 Make call.....	111
7.1.6 Answer.....	112
7.1.7 Hang up call.....	113
7.1.8 Record audio message.....	113
7.1.9 Get call tag.....	114

7.1.10 DTMF tones.....	114
7.1.11 Tag call.....	115
7.2 User Management Actions.....	116
7.2.1 Get user.....	116
7.2.2 Search users.....	117
7.2.3 User presence.....	118
7.3 Developer Actions.....	119
7.3.1 Condition.....	121
7.3.2 Set variables.....	122
7.3.3 Http request.....	122
7.3.4 Log.....	123
7.3.5 Terminate.....	123
7.3.6 For each.....	123
7.3.7 Go to.....	124
7.3.8 Break.....	124
7.3.9 Delay.....	124
7.3.10 Continue.....	124
7.3.11 Bookmark.....	124
7.3.12 Open session.....	125
7.3.13 JavaScript.....	125
7.3.14 Cloudlink subscribe.....	126
7.3.15 Cloudlink unsubscribe.....	127
7.3.16 Parallel.....	127
7.3.17 Try-Catch.....	127
7.3.18 While.....	128
7.4 Routing Actions.....	128
7.4.1 Schedule.....	129
7.4.2 Route interaction.....	129
7.4.3 Emergency mode.....	130
7.4.4 Phone number lookup.....	131
7.5 MCX Actions.....	132
7.5.1 MCX get employee state.....	133
7.5.2 MCX change employee state.....	134
7.5.3 MCX get queue state.....	135
7.5.4 MCX port state.....	135
7.5.5 MCX route open media.....	136
7.5.6 MCX send data.....	137
7.5.7 MCX callback.....	137
7.6 Messaging Actions.....	138
7.6.1 Create chat stream.....	141
7.6.2 Send chat message.....	142
7.6.3 Ask.....	143
7.6.4 Add participant.....	144
7.6.5 Tag chat.....	144
7.6.6 Delete chat.....	145
7.6.7 Create chat transcript.....	146
7.6.8 Create or update attachment.....	146
7.6.9 Get chat tag.....	147
7.6.10 Get chats with tag.....	148
7.6.11 Delete chat tag.....	148
7.6.12 Delete participant.....	149
7.6.13 Get participants.....	149
7.6.14 Add endpoint participant.....	150
7.6.15 Send command.....	150
7.6.16 Send whisper.....	151
7.7 Integrations Actions.....	152

7.7.1 JIRA Cloud Actions.....	152
7.7.2 Microsoft Graph Actions.....	155
7.7.3 Salesforce Actions.....	170
7.7.4 CM.com Actions.....	174
7.7.5 Github Actions.....	176
7.7.6 MiTeams Meeting Actions.....	180
7.7.7 Other Integration Actions.....	183
7.8 Search Only Actions.....	193
7.8.1 Detect sentiment.....	196
7.8.2 Translate.....	197
7.8.3 Send message - 69xx phone.....	197
7.8.4 Persisted variables.....	198
7.8.5 Weather.....	199
7.8.6 Publish notification.....	200
7.8.7 Slack send.....	201
7.8.8 MiCollab phone number parser.....	201
7.8.9 Create contact.....	202
7.8.10 Delete contact.....	203
7.8.11 Tag user.....	203
7.8.12 Add record MiVB Form.....	204
7.8.13 Delete record MiVB Form.....	204
7.8.14 Get MiVB form data.....	205
7.8.15 Update record MiVB Form.....	206
7.8.16 AI CloudLink Chat Bot.....	206
7.8.17 AI Voice Bot.....	207

8 List of Triggers.....209

8.1 Messaging Triggers.....	209
8.1.1 Chat message received.....	210
8.1.2 Chat created.....	211
8.1.3 CC Messenger.....	211
8.2 Media Triggers.....	213
8.2.1 Endpoint ringing.....	213
8.2.2 Extension state changed.....	214
8.3 User Management Triggers.....	215
8.3.1 Presence changed.....	215
8.3.2 User changed.....	216
8.3.3 User deactivation/deletion.....	216
8.3.4 User tag change - add, update and delete.....	216
8.4 Integrations Triggers.....	217
8.4.1 Twilio incoming messages.....	217
8.4.2 CM.com incoming messages.....	219
8.4.3 Zoom trigger.....	220
8.5 Developer Triggers.....	221
8.5.1 Webhook.....	221
8.5.2 Sign In App notification.....	222
8.5.3 Schedule.....	222
8.5.4 SIP DECT notification.....	223
8.6 Search Only Triggers.....	224
8.6.1 Slack webhook.....	224
8.6.2 Presence source changed.....	225
8.6.3 Github issues.....	226
8.6.4 VCCS Trigger.....	227

9 Integrations..... 228

9.1 Anthropic Integration (Premier).....	228
9.1.1 Create Anthropic Integration.....	228
9.1.2 Add an Anthropic activity.....	230
9.1.3 Anthropic Claude Use Case.....	231
9.2 CM.com Integration (Premier).....	232
9.2.1 Create CM.com Integration.....	232
9.2.2 Add a CM.com Send Message activity.....	233
9.3 Edge Signal Integration (Premier).....	236
9.3.1 Create Edge Signal Integration.....	236
9.3.2 Add an Edge Signal Trigger.....	237
9.3.3 Add an Edge Signal Activity.....	238
9.4 GitHub Integration (Essential).....	239
9.4.1 GitHub Requirements.....	240
9.4.2 Create GitHub Integration.....	240
9.4.3 Use the GitHub API in Workflow.....	241
9.5 Google Integration (Essential).....	243
9.5.1 Prerequisites.....	243
9.5.2 Create Google Integration.....	243
9.5.3 Use Google APIs in Workflow.....	247
9.5.4 Use Google with Workflow Activities.....	248
9.6 Google Gemini Integration (Premier).....	251
9.6.1 Create Google Gemini Integration.....	251
9.6.2 Add a Google Gemini Activity.....	253
9.6.3 Google Gemini Use Case.....	254
9.7 HTTP Integration (Premier).....	255
9.7.1 Create HTTP Integration.....	255
9.7.2 Add an HTTP Request Activity.....	258
9.7.3 Business Use case.....	259
9.8 Jira Cloud Integration (Premier).....	264
9.8.1 Create Jira Cloud Integration.....	264
9.8.2 Add Jira Cloud Activities.....	266
9.9 Microsoft Azure Integration (Premier).....	267
9.9.1 Create Microsoft Azure Bot Integration.....	267
9.10 Microsoft Graph Integration (Premier).....	268
9.10.1 Create Microsoft Graph Integration.....	269
9.10.2 Add a Graph Connection activity.....	274
9.10.3 Retrieve the Teams ID.....	275
9.10.4 Permissions Required.....	277
9.11 OpenAI Assistants Integration (Premier).....	282
9.11.1 Prerequisites.....	282
9.11.2 Set up your OpenAI Assistants.....	283
9.11.3 Create OpenAI Integration.....	284
9.11.4 Configure the media types.....	286
9.12 Outlook Integration (Essential).....	288
9.12.1 Create a new application/registration within Microsoft Azure (Entra).....	288
9.12.2 Create Outlook Integration.....	290
9.12.3 Add an Office 365 Outlook - Send Email activity.....	293
9.13 Salesforce Integration (Premier).....	296
9.13.1 Setup an OAuth flow.....	296
9.13.2 Create Salesforce Integration.....	297
9.13.3 Add a Salesforce activity.....	299
9.14 Twilio Integration (Premier).....	300

9.14.1 Create Twilio Integration.....	300
9.14.2 Add a Twilio send message activity.....	302
9.15 Whispeak Integration (Premier).....	304
9.15.1 Create Whispeak Integration.....	304
9.15.2 Add a Whispeak Create Signature activity.....	305
9.15.3 Add a Whispeak Authenticate activity.....	307
9.15.4 Add a Whispeak Delete Signature activity.....	309
9.15.5 Sample Use Case.....	311
9.16 Zoom Integration (Essential).....	312
9.16.1 Prerequisites.....	312
9.16.2 Create a New Application in Zoom.....	312
9.16.3 Create Zoom Integration.....	315
9.16.4 Configure the Trigger Notification in the Zoom App.....	318
9.16.5 Add a Zoom Trigger.....	322
9.16.6 Use the Zoom API in Workflow.....	324

10 Best Practices..... 327

10.1 How to configure the Dial Menu timeout settings and destination.....	327
10.2 How to configure the Dial Menu transfer to a voicemail box.....	328
10.3 How to configure the Replay Dial Menu Greeting.....	329
10.4 How to configure the hang up on caller activity.....	330
10.5 How to configure the business opening hours for public holidays.....	330
10.6 How to reuse activities in many flows.....	332

11 Security Guidelines..... 333

11.1 Securing Data - Configuring and Executing Flows.....	333
11.2 Configuration of Third-party Integrations.....	336
11.2.1 Storing of Client secrets/tokens/API Keys.....	336
11.3 Rate Limits.....	337
11.4 Data Retention Schedule.....	338
11.5 Compliance.....	339

12 Troubleshooting..... 340

12.1 Variable Scope.....	340
12.2 Flow Expression Calculations.....	341
12.3 Handling a Space in a JSON Parameter.....	343
12.4 Variables are not being replaced with the correct value.....	343
12.5 String Replacement Syntax.....	343
12.6 Integrations.....	344

This chapter contains the following sections:

- [Purpose](#)
- [Intended Audience](#)
- [Acronyms and Abbreviations](#)
- [Terminologies](#)
- [Licensing](#)
- [Related Documents](#)

1.1 Purpose

This guide provides an overview of Mitel Workflow Studio and covers various information about implementing, configuring, and troubleshooting workflows.

1.2 Intended Audience

This guide is intended for Mitel Workflow Studio administrators.

1.3 Acronyms and Abbreviations

The following acronyms and abbreviations are used in this guide:

Table 1: Acronyms and Abbreviations

Acronyms and Abbreviations	Description
AI	Artificial Intelligence
API	Application Programming Interface
AWS	Amazon Web Services
GUID	Globally Unique Identifier
HTTP	Hypertext Transfer Protocol

Acronyms and Abbreviations	Description
IVR	Interactive Voice Response
JSON	JavaScript Object Notation
MiCC	MiContact Center
MiVB	MiVoice Business
MiVO400	MiVoice Office 400
MiV5000	MiVoice 5000
PBX	Private Branch eXchange
SVG	Scalable Vector Graphic
SWA	Software Assurance
URI	Uniform Resource Identifier
UTC	Coordinated Universal Time

1.4 Terminologies

The following terms are used in this guide:

Table 2: Basic Terms

Basic Terms	Description
Action	<p>A configurable task within a flow, which can be a Primitive or Composite.</p> <ul style="list-style-type: none"> • Primitive actions provide basic functions such as Set Variables, Conditions, or HTTP requests. • Composite actions are built using primitive actions and provide more complex operations.

Basic Terms	Description
Activity	A generic term referring to any trigger, flow, action, or anything that 'runs' or 'does something'.
Flow	A group of configurable actions that initiate with a trigger and are executed from left to right. It includes defined inputs, throughputs, and outputs that can be used to pass information, which can be retrieved once the flow is completed. It also includes local and shared variables that are available throughout the flow.
Node	A general term referring to any object within a flow.
Template	Pre-built globally available flows that can be cloned locally into the admin's account. Once cloned, these templates become flows that can be configured to work within the admin's environment.
Trigger	The first node in a flow. It initiates a flow and specifies the conditions under which it should be executed or triggered. It can be either Composite or Primitive and is available globally.
Worker	A unique running instance of a flow with its own GUID.

1.5 Licensing

Mitel Workflow Studio does not need a license; it is available to all accounts with a configured CloudLink Gateway and valid Software Assurance (SWA). The minimum required level of SWA is Advantage/Standard SWA. If the SWA has expired, Mitel Workflow Studio cannot be enabled.

1.6 Related Documents

The related documents are as follows:

- Mitel Administration User Guide
- Mitel Workflow Studio System Engineering Guide
- Release Notes

Mitel Workflow Studio – An Overview

2

This chapter contains the following sections:

- [Sample Use Cases](#)
- [Limitations](#)

Mitel Workflow Studio is a web-based user interface for designing and configuring workflows, executed when a specific trigger condition is met. These flows can automate various functions within the Mitel ecosystem (such as IVR call or chat flows) and serve as an Integration Platform-as-a-Service (IPaaS) to connect business systems and process data.

The workflow is one of the various microservices within the CloudLink platform that runs on Amazon Web Services. Unlike the other CloudLink microservices, primarily accessed through RESTful web service (API), Workflow is accessed mainly through Mitel Workflow Studio.

2.1 Sample Use Cases

Many Mitel products already use Workflow to power certain features and functions, such as **Mitel Voice Assist**, which exists almost entirely within Workflow, and **MiContact Center Business**, which leverages Workflow for chat routing and AI functionality. Using Workflow, you can use an existing library of customizable prebuilt flows or start from scratch using a wide variety of available actions that allow you to create flows limited only by your imagination (and the computational power of AWS).

2.2 Limitations

As a web-based solution, Workflow is limited by restrictions on accessing on-premises systems via the Internet. Support for some call control functionality is available for on-premises systems (MiVB, MiVO400, MiV5000, and MX-ONE) through a supported CloudLink Gateway.

This chapter contains the following sections:

- [Workflow Integrations](#)
- [Flows, Actions, and Triggers](#)
- [Workflow Dashboard Usage Reporting](#)
- [License Usage Notifications and Overages](#)
- [Customer Responsibilities](#)
- [Subscription Expiry](#)

Workflow Studio features and usage are subject to licensing tiers, which define access to Integrations, Flows, Actions, and Triggers. These features are categorized under **Premier** and **Essential** licenses.

3.1 Workflow Integrations

Workflow integration's are divided into two categories: **Premier** and **Essential**. Your current license determines the availability of each integration.

- **Premier Integrations** offer enhanced capabilities and connect to advanced third-party services. Workflow Studio ships with a predefined set of Premier integrations that provide extended functionality, enterprise system connectivity, and advanced automation capabilities. These integrations include platforms such as Salesforce, Microsoft Graph, Twilio, Zoom, OpenAI, and others.
- **Essential Integrations** provide core integration functionality for commonly used platforms.

Users can only create integration connections for integrations included in their license tier. The interface will disable non-licensed integrations.

Within Workflow Studio, each integration is clearly marked with a **Premier** or **Essential** tag to indicate its licensing classification. This helps administrators and users easily identify the required for each integration.

3.2 Flows, Actions, and Triggers

All workflow components—**flows**, **actions**, and **triggers**—are similarly categorized as **Premier** or **Essential**.

- Users may only create flows, actions, and triggers that are included in their current license tier.
- Newly created activities are automatically tagged based on their contents. If an activity includes a Premier-level action or trigger, the entire activity will be classified as **Premier**. Otherwise, it will be tagged as **Essential**.
- Flows, actions, and triggers are individually tagged to reflect their licensing level.

Each flow, action, and trigger in the Workflow Studio interface is clearly marked with a **Premier** or **Essential** tag to indicate its licensing classification. This helps administrators and users easily identify the level of access required for each activity.

Refer to the **License Type** column under the [List of Actions/List of Triggers](#) chapter to understand the license classification of each action/trigger.

3.3 Workflow Dashboard Usage Reporting

The **Workflow Dashboard** includes a real-time overview of workflow usage, including:

- A display of the **Top 5 running workflows** and their execution count over the current billing period.
- A usage summary indicating how many **Premier** and **Essential** workflows have been executed in the current billing period.
- Progress indicators that show how close the account is to its monthly usage limits based on the current license.

The license usage total in percentage and the monthly allowed execution limit are displayed.

These insights help administrators manage capacity, monitor utilization, and forecast licensing needs.

3.4 License Usage Notifications and Overages

Workflow Studio monitors the number of **flow executions** against the limits defined by your account's **Premier** and **Essential** license tiers.

As usage approaches or exceeds these limits, Workflow Studio will automatically notify the CloudLink Support contacts to ensure proactive awareness and management:

- **At 80% of usage:** A warning email is sent to all support contacts defined in the **Mitel Administration** portal, notifying them that execution limits are being approached.
- **At 100% of usage:** A second email is sent indicating that the account has reached its licensed capacity.
- **At 120% of usage:** A critical email is sent indicating that the account has significantly exceeded its licensed execution limits.



Note:

Once the account exceeds 120% usage, you will not be allowed to create, save, and clone workflows.

These alerts help prevent service degradation and allow time to take action.

3.5 Customer Responsibilities

Upon receiving these notifications, customers are responsible for addressing the situation by either:

- Reducing the frequency and volume of workflow executions to bring usage back within allowed limits, or

- Purchasing additional Premier and/or Essential licensing to accommodate current and future usage.

It is important to monitor usage through the Workflow Dashboard, which provides up-to-date visibility into the number of executions and license consumption trends.

3.6 Subscription Expiry

Access to Workflow Studio is dependent on a valid Premier and/or Essential subscription. If these subscriptions expire, the following behavior will occur:

- Access to Workflow Studio will be revoked. Users will no longer be able to access the application.
- When attempting to log into Workflow Studio, users will see a banner notification indicating that their license has expired.
- After the banner is displayed, users will automatically be logged out of the application.

To regain access, customers must renew their Workflow Studio subscription through the appropriate Mitel licensing channels.

This chapter contains the following sections:

- [Integrating Workflow Studio](#)
- [Migrating Voice Assist to Workflow Studio](#)
- [Logging In](#)
- [Reporting a problem](#)
- [Logging Out](#)
- [Application Environment](#)

Workflow Studio is available in the **Mitel Administration Accounts** console and requires a dedicated CloudLink User with the ACCOUNT_ADMIN role (or higher). Partners can access their customer accounts by assuming a role in the customer account.

Currently, **Mitel Workflow Studio** is only supported on Google Chrome.

Refer to the [Mitel Administration User Guide](#) for more details.

4.1 Integrating Workflow Studio

This topic provides step-by-step instructions on how to integrate the Workflow Studio application into the **Mitel Administration** account.

Perform the following steps to integrate **Workflow Studio**, if it is not integrated into the **Mitel Administration** account:

1. Log in to **Mitel Administration** using the URL (<https://accounts.mitel.io>).
2. Enter a valid username and password.

The **Mitel Administration Dashboard** is displayed upon successful login.

3. Click the **Integrations & Apps** option from the navigation menu.
4. Enable the toggle button associated with the **Workflow Studio** integration in the **Integrations** panel.

The **Workflow Studio** icon is now available under the **Application Launcher** icon at the top of the page.

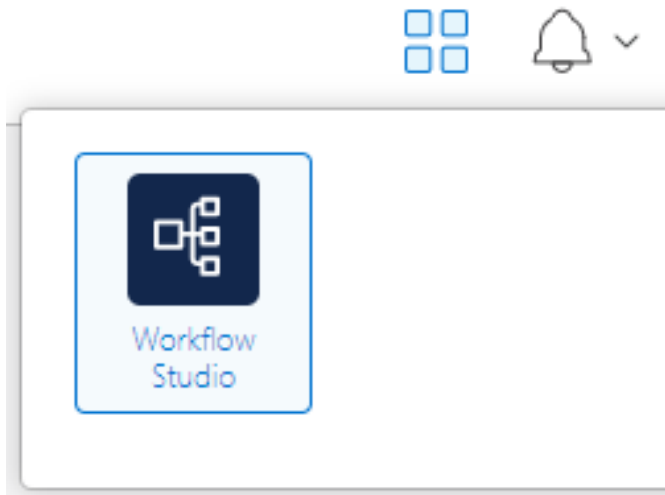


Figure 1: Application Launcher

4.2 Migrating Voice Assist to Workflow Studio

This topic provides information about the transition from Voice Assist to the new Mitel Workflow Studio.

Migration Overview

Mitel Voice Assist will be phased out in the future and replaced with Mitel Workflow Studio. Using GenAI-powered voice agents and CRM integrations, you have more options to enhance workflows and make them smarter. Mitel Workflow Studio provides many new workflow triggers and actions that are not limited to telephony/voice only. Workflow interactions can also be triggered by digital channels like chat, email, webhooks, and SMS, which allow new workflow automation in combination with Mitel Voice platforms and UC/ CX applications.

The table below highlights how users previously configured flows in Voice Assist compared to the new experience in Workflow Studio.

Table 3: Product Change Comparison

Feature	Voice Assist	Workflow Studio
Accessing the application	Log in to Mitel Administration Portal > Voice Assist > Routing	Log in to Mitel Administration Portal > Workflow Studio > Flows
Flow execution counts	Routing page	Dashboard page with a detailed list of flow executions
Creating Flows	Routing page > Add Route	Flows page > Create Flow

Feature	Voice Assist	Workflow Studio
Deleting Flows	Click the ... on the flow name row and click Delete	Flows page > Click the Delete icon next to the flow
Duplicating Flows	Click the ... on the flow name row and click Duplicate	Flows page > Click the Clone Button on the flow name
Activity Wheel	Limited	More activities are available across different modalities
Numbers View	Voice Assist -> Numbers	This view is obsolete; filter the flows list by Endpoint Trigger to view flows that are voice-specific.
Integrations	Limited	More integrations

4.3 Logging In

This topic provides step-by-step instructions on how to log in to the application.

Mitel Workflow Studio can be logged in using the following options:

Direct Option:

1. Log in to **Mitel Workflow Studio** using the URL (<https://workflow.mitel.io/>).
2. Enter a valid username and password.
3. Click **Next**.

The **Mitel Workflow Studio Dashboard** is displayed.

Through Mitel Administration:

1. Log in to **Mitel Administration** using the URL (<https://accounts.mitel.io/>).
2. Enter a valid username and password.
3. Click **Next**.

The **Mitel Administration Dashboard** is displayed upon successful login.

4. In the **Application Launcher** icon, click on the **Workflow Studio** icon to launch the application.

The **Mitel Workflow Studio Dashboard** is displayed.

4.4 Reporting a problem

To report a problem, perform the following steps:

1. On the **Workflow Studio** home page, click on the username or avatar logged into the application.

The **User Profile** dropdown list is displayed.

2. Click **Report a problem**.

The **Issues & Feedback** screen is displayed.

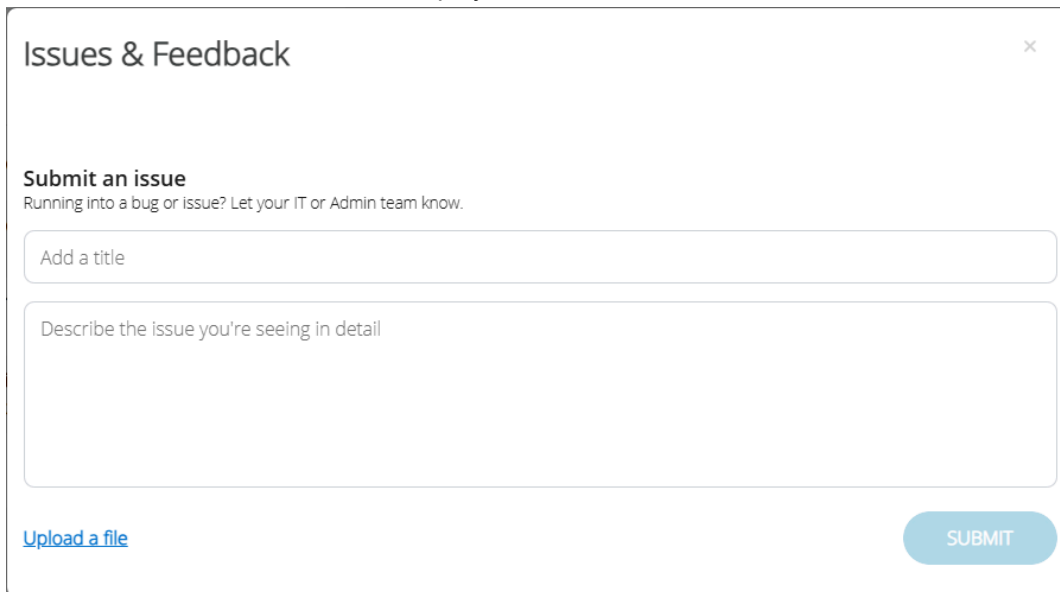
The screenshot shows a web form titled "Issues & Feedback" with a close button (X) in the top right corner. Below the title is a section "Submit an issue" with the text "Running into a bug or issue? Let your IT or Admin team know." There are two input fields: a text field labeled "Add a title" and a larger text area labeled "Describe the issue you're seeing in detail". At the bottom left is a link "Upload a file" and at the bottom right is a blue "SUBMIT" button.

Figure 2: Issues & Feedback

3. Enter the title and brief description of the issue in the respective fields.
4. In the **Upload a file** field, search and select the file related to the issue. For example, attach a screenshot or Console log from your browser that provides additional information about the issue.
5. Click **SUBMIT**.

The Issue or Feedback is reported to the support contacts of your CL account.

Note:

- While submitting a problem report, provide as much information as possible. This will help troubleshoot your problem quickly. Specific details are needed to correctly investigate and resolve an issue.
- Date, time, device information, and logs are collected as part of your problem report.
- After you report an issue, the Support Contact for your account will receive an email notifying them that you have reported a problem. The Support Contact, while investigating the issue, might reach out to you for additional information.
- If your Support Contact needs assistance to resolve your issue, they will escalate the report through their standard support procedures.
- You can follow up directly with your Support Contact for updates about your report.

4.5 Logging Out

This topic provides step-by-step instructions on how to log out of the application.

1. On the **Workflow Studio** home page, click on the username or avatar logged into the application.

The **User Profile** dropdown list is displayed.

2. Click **Logout** to log out from the application.

The application logs out.

4.6 Application Environment

This topic describes the various components of the application environment.

The **Workflow Studio – Home page** appears after successful login, depending on the user's privileges.

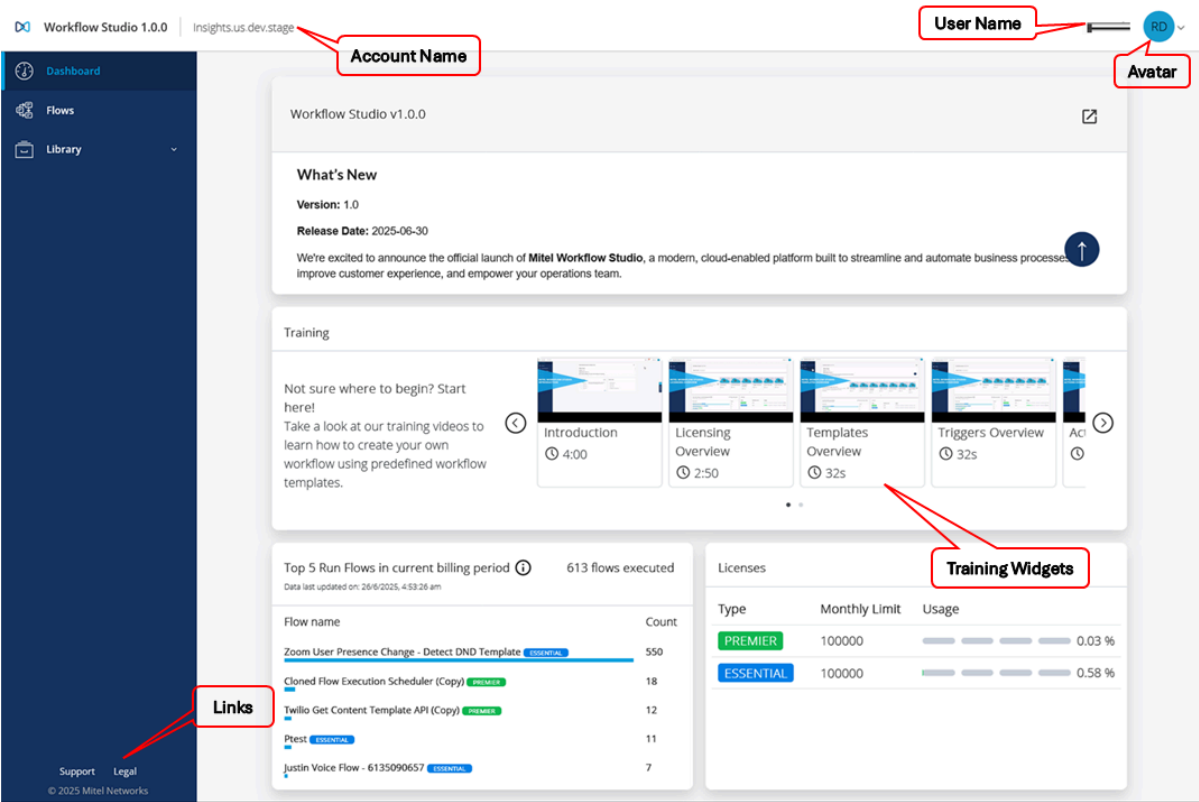


Figure 3: Workflow Studio – Home page

The following table describes the user interface elements on the home page.

Table 4: Workflow Studio – Element Description

UI Elements	Description
User Name	Displays the name of the logged-in user as in the Mitel Administration portal.
Avatar	Displays a short representation or image of the logged-in user.
Account Name	Displays the customer account name added by the Mitel Partner.
Dashboard	Displays the widgets to view the training videos for predefined templates and a list of the top five run flows in the current billing period.
Flows	Displays the list of workflows accessible to the user and allows the user to create, edit, import, clone, and delete workflows.
Library	Displays the list of available actions, triggers, templates, and integrations for the account.

UI Elements	Description
Links	<p>The following links appear at the bottom of Workflow Studio:</p> <ul style="list-style-type: none">• Support – Access to the online help.• Legal – Access to the application end-user license agreement.

Understanding Workflow Studio

5

This chapter contains the following sections:

- [Dashboard](#)
- [Flows](#)
- [Library](#)

This chapter provides an overview of navigating and utilizing the Workflow Studio application effectively.

Workflow Studio is primarily accessed through the following navigation menus:

1. Dashboard
2. Flows
3. Library

5.1 Dashboard

The **Dashboard** provides various widgets to view the release notes, training videos, top 5 running flows, and licenses.

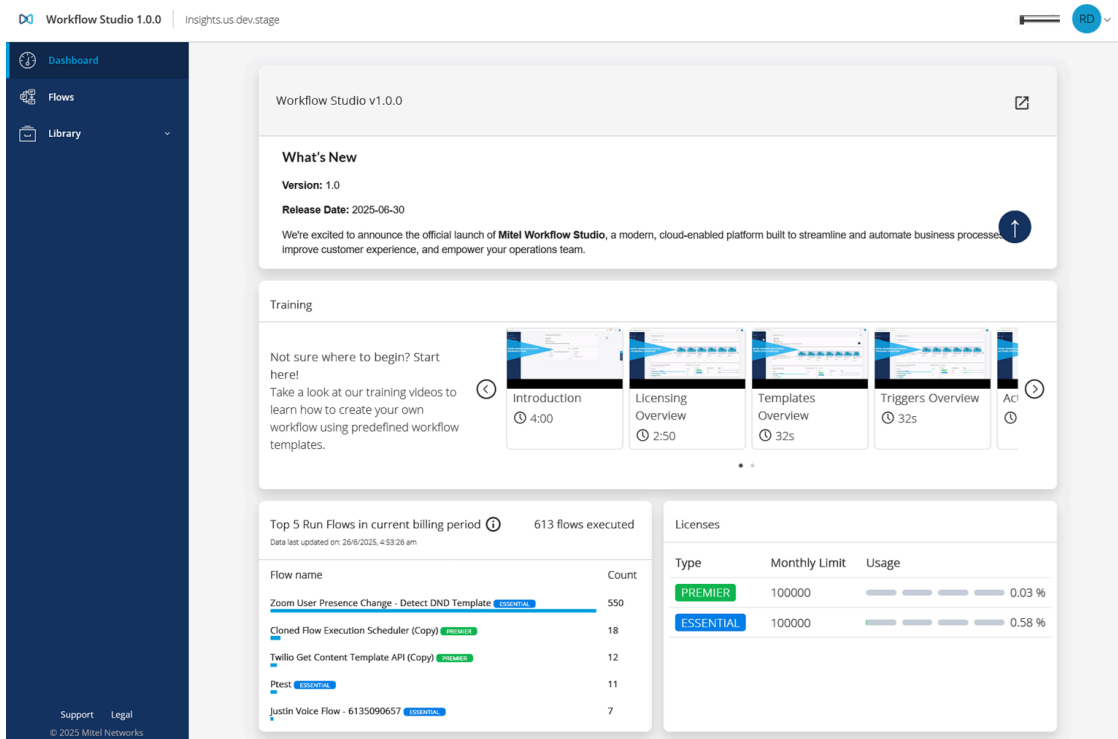


Figure 4: Dashboard

It consists of several sections:

- **What's New** – Displays the release notes added in the current release.
- **Training** – Displays the templates and training videos of the flows on various use cases.
- **Top 5 Run Flows in current billing period** - Displays the total number of flows executed on the account in the current billing period and the total number of flows allowed for the account during the current monthly billing period.
- **Licenses** - Displays the license type, total number of executable licenses for the current period, and the usage percentage of each license type.

5.1.1 What's New

You can view the list of new features added, existing features modified/removed, and bugs fixed in the current release version, along with the release date.

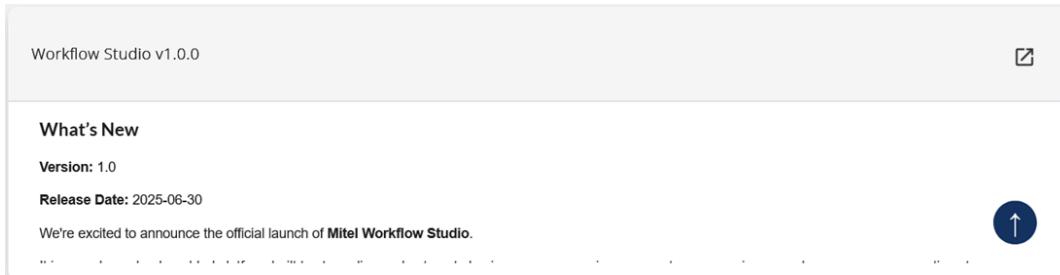



Figure 5: What's New

The following actions can be performed using the **Release Notes** section.

- Click on the **What's New in Workflow Studio** hyperlink to view the parent topic of the release notes.
- Click  icon to open the complete release notes in a new tab.

5.1.2 Training

You can use the training videos to understand the **Workflow Studio** product and its functionalities.

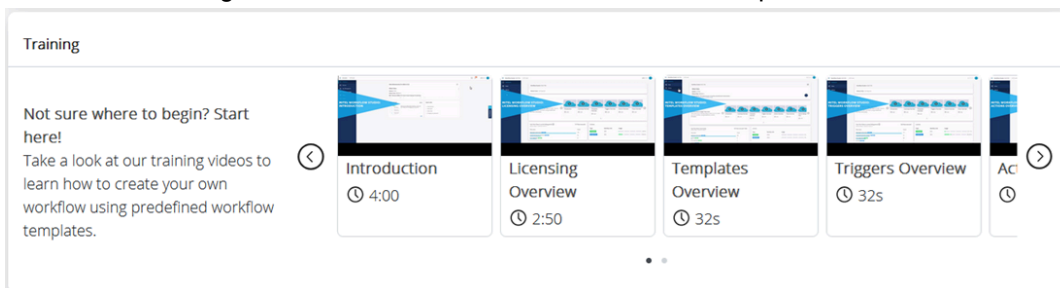


Figure 6: Training

In the **Training** widget, click on the respective tile to launch the training video in a new browser tab.

5.1.3 Top 5 Run Flows in current billing period

You can view the total number of flows executed on the account in the current billing period and the total number of flows allowed for the account monthly under the **Top 5 Run Flows in current billing period** section.

This data is refreshed once every 24 hours. The timestamp of the most recent update is displayed at the top of the table. If a workflow is renamed after it has been executed, the original name will continue to appear until the next day's update.

For example, if a workflow is initially named "Workflow 1" and executed twice, then renamed to "Workflow 2" and executed three more times, the table will show "Workflow 1" with a total of five executions the next day. The updated name "Workflow 2" will be reflected in the subsequent day's data.

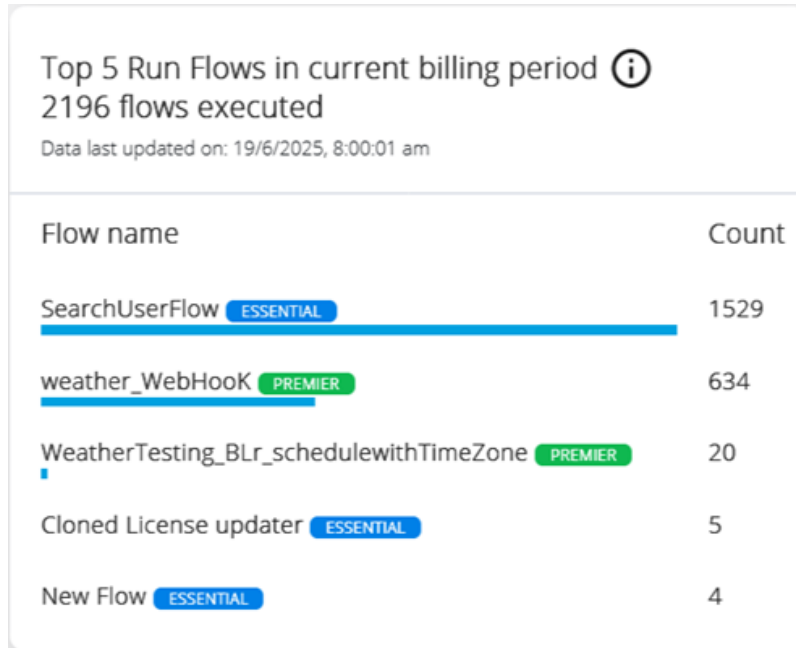


Figure 7: Top 5 Run Flows in current billing period

The following details can be displayed in the **Top 5 Run Flows in current billing period** section:

- the total number of flows executed on the account in the current billing period.
- the last updated date and time of the displayed data (since this is not real-time and is updated periodically).
- the names of the top five most executed workflows in the current billing period, the type of license used (Essential or Premier), and their execution count.

Note:

- If the user exceeds 80% and 100% of the usage limit, the email notifications are sent.
- If the user exceeds 120% of the allowed number of flows, it will not be possible to create, clone, or save new activities within Workflow Studio.

5.1.4 Licenses

The **Licenses** widget provides detailed information on the two available Workflow Studio license types: **Premier** and **Essential**. It enables users to track license consumption during the current billing cycle. The

billing period's start and end dates can be found in the **Top 5 Run Flows in current billing period** section of the Dashboard.

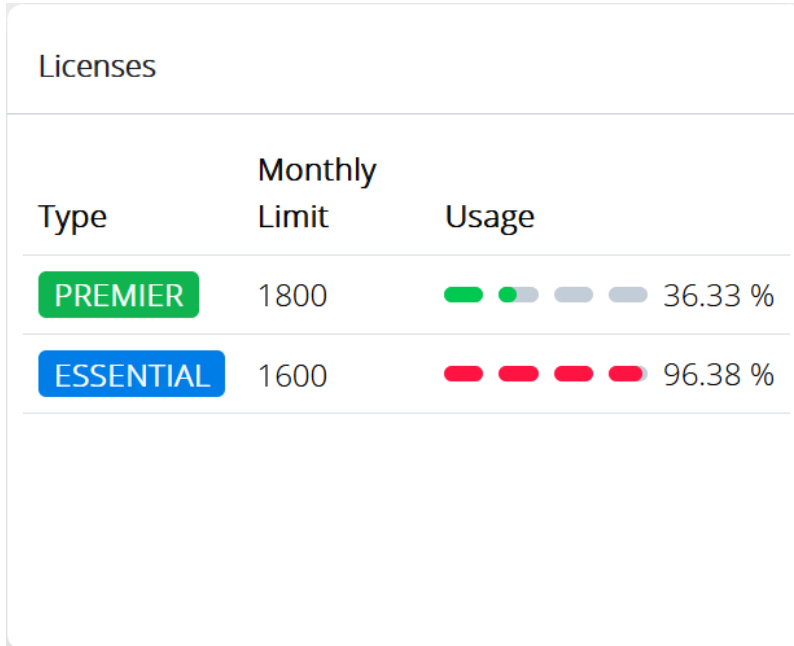


Figure 8: Licenses

When either license type is active, the dashboard displays both the total number of executable licenses for the current period and the percentage of usage for each type. To indicate the current usage, the graph line is color-coded as:

- **Green:** Usage is below 50% of the monthly quota.
- **Yellow:** Usage is between 50 to 75%.
- **Red:** Usage exceeds 75%.

If the usage exceeds the monthly limit, the existing workflow will continue to execute normally. But you will be restricted from adding/modifying the workflow unless additional licenses are purchased.

5.2 Flows

Flows automate business processes through a structured sequence of triggers, actions, and conditions. They consist of interconnected steps where triggers are set, actions are performed, data is manipulated, and decisions are made based on defined conditions. You can create workflows by adding the triggers and actions onto a Workflow Canvas and configuring them to achieve desired outcomes. Workflows have logical branches and conditional paths to accommodate diverse scenarios. Organizations use workflows to streamline operations, enhance productivity, and maintain consistency in task execution.

The **Flows** page displays a list of workflows accessible to the user and information about who created them, when they were last updated, and who modified them.

You can access the **Flows** page by clicking **Flows** on the left menu.

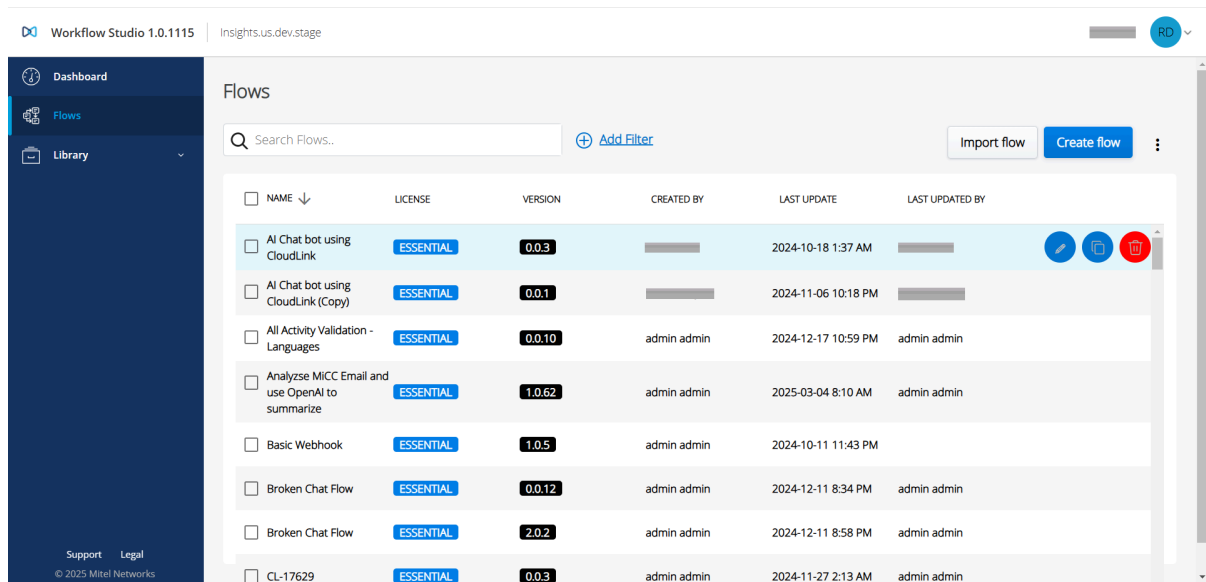


Figure 9: Flows

The following actions can be performed under the **Flows** section:

- **Create a flow**
- **Edit a flow**
- **Import a flow**
- **Clone a flow**
- **Delete a single or multiple flow(s)**

5.2.1 Create a flow

This topic provides step-by-step instructions for creating a new workflow. Follow the training videos in the **Dashboard** section to create a new workflow.

1. On the **Workflow Studio** home page, click **Flows**.

The **Flows** page is displayed.

2. On the **Flows** page, click **Create flow**.

The **Create flow** page is displayed.

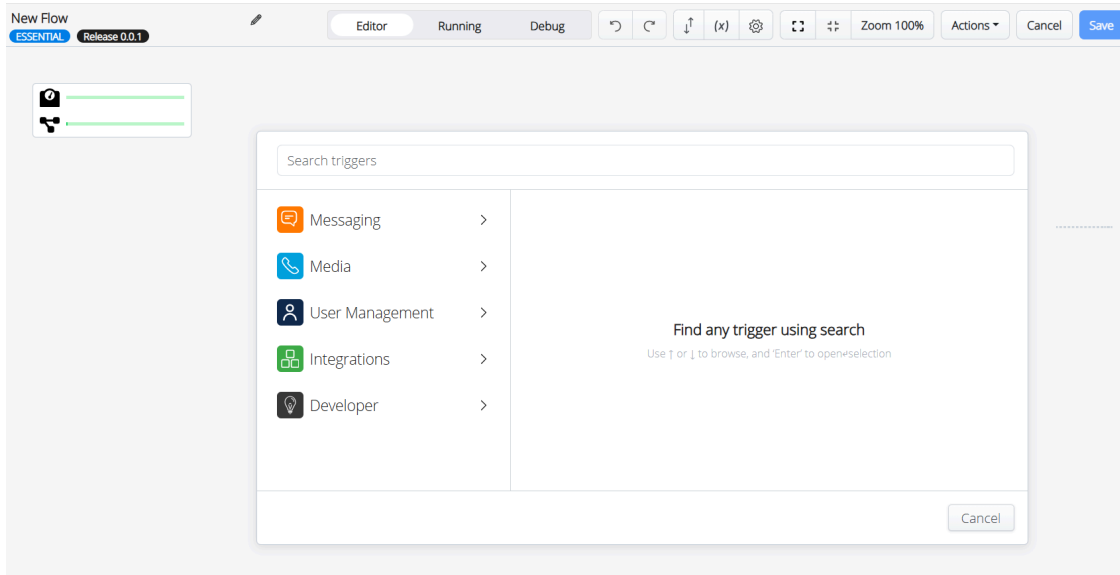


Figure 10: Create flow

3. Enter the trigger name in **Search activities** or select the trigger under the trigger types.

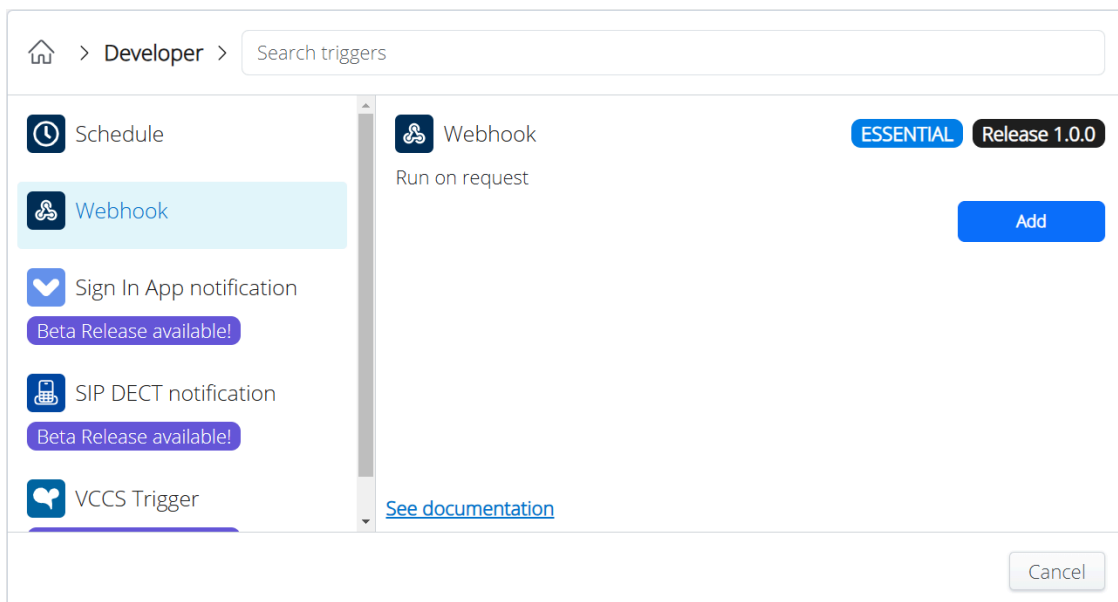


Figure 11: Add Trigger

4. Click **Add**.

The **Trigger Parameters – Advanced** screen is displayed.

The screenshot displays the 'Trigger Parameters – Advanced' configuration window. At the top, it shows 'New Flow' with a 'Webhook' trigger type, an 'ESSENTIAL' badge, and 'Release 1.0.0'. A warning message states: 'It is recommended to set a Client Secret so external services may access this flow'. An 'Add annotations' button is present. Below this, a lock icon and text prompt the user to 'Insecure route. Please generate a secret:'. To the right are icons for deleting, editing, and refreshing the secret. The 'Invoke Instructions' section has a copy icon. The main area is a text editor showing a POST request with headers 'Content-Type: application/json' and 'x-mitel-shared-secret: YOUR_CLIENT_SECRET', and an empty body. A toggle switch for 'Execute synchronously' is at the bottom left. The bottom toolbar contains icons for delete, copy, paste, undo, redo, and help, followed by 'Cancel' and 'OK' buttons.

Figure 12: Trigger Parameters - Advanced

5. Enter the required parameters in the **Trigger Parameters – Advanced**.

6. Click **Advanced**.

The **Trigger Parameters – Classic** screen is displayed.

The screenshot shows the 'Trigger Parameters – Classic' dialog box. At the top, it says 'New Flow' with a 'Webhook' icon and 'Release 1.0.0'. Below this, it says 'Webhook' and 'Run on request'. A yellow warning icon and text state: 'It is recommended to set a Client Secret so external services may access this flow'. There is a yellow 'Add annotations' button. Below this, the 'ID' field contains 'Webhook1'. At the bottom, there are three buttons: '+ Input', '+ Throughput', and '+ Output'. The footer contains a toolbar with icons for delete, copy, paste, undo, redo, and help, along with 'Cancel' and 'OK' buttons.


Figure 13: Trigger Parameters – Classic

7. Provide the following details in the **Trigger Parameters – Classic** screen.

- a. ID
- b. Input
- c. Throughput
- d. Output

8. Click **OK**.

The **Trigger** is added to the flow.

9. Click  icon to add the next activity to the flow.

The **Search Activities** wheel is displayed.

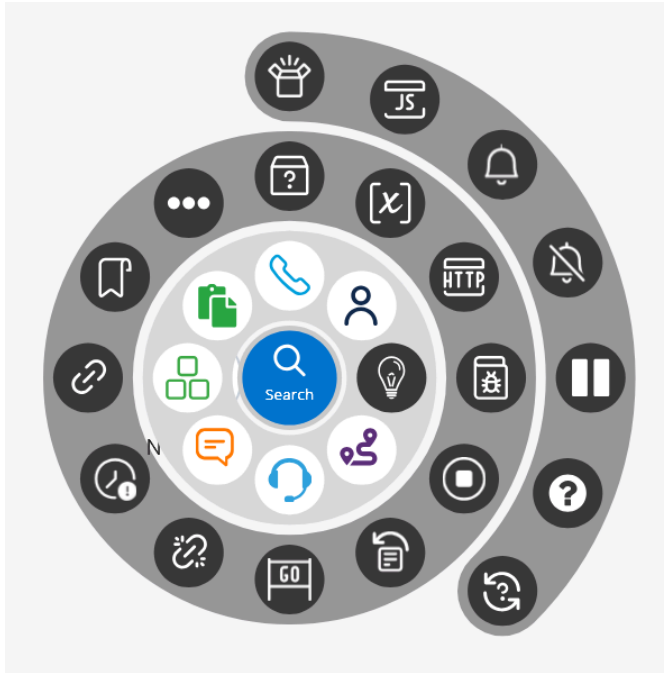


Figure 14: Search Activities

- 10.** Search and select the required activity from the wheel.

The **Activity Parameters** screen is displayed.

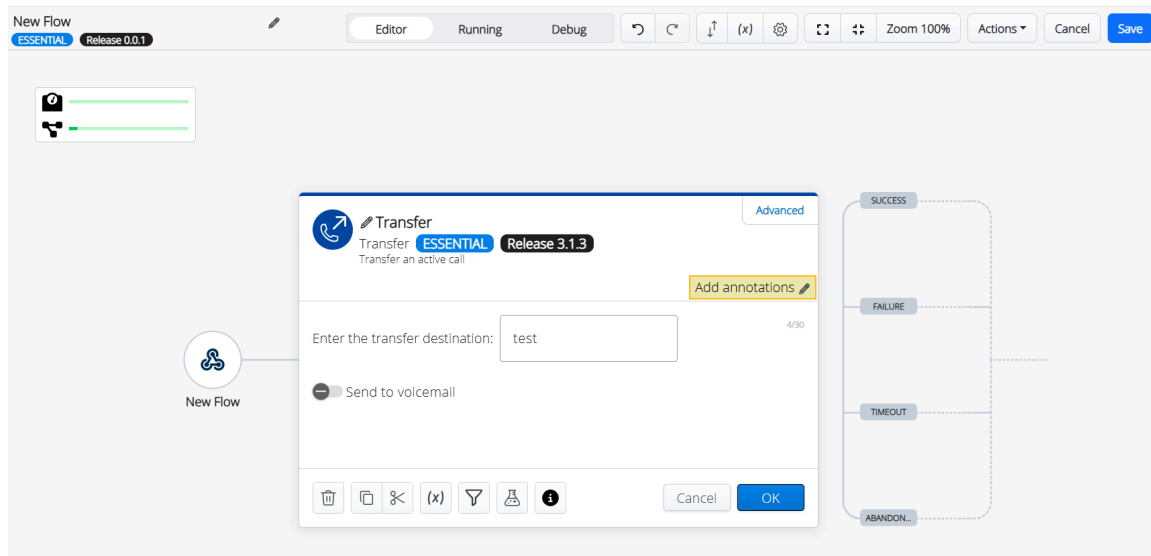


Figure 15: Activity Parameters

11. Provide the required details in the **Activity Parameters** screen.
12. Click **OK**.

13. Follow the above steps to add all the required activities for the flow.

The **Completed flow** screen is displayed with the trigger and all the required actions.

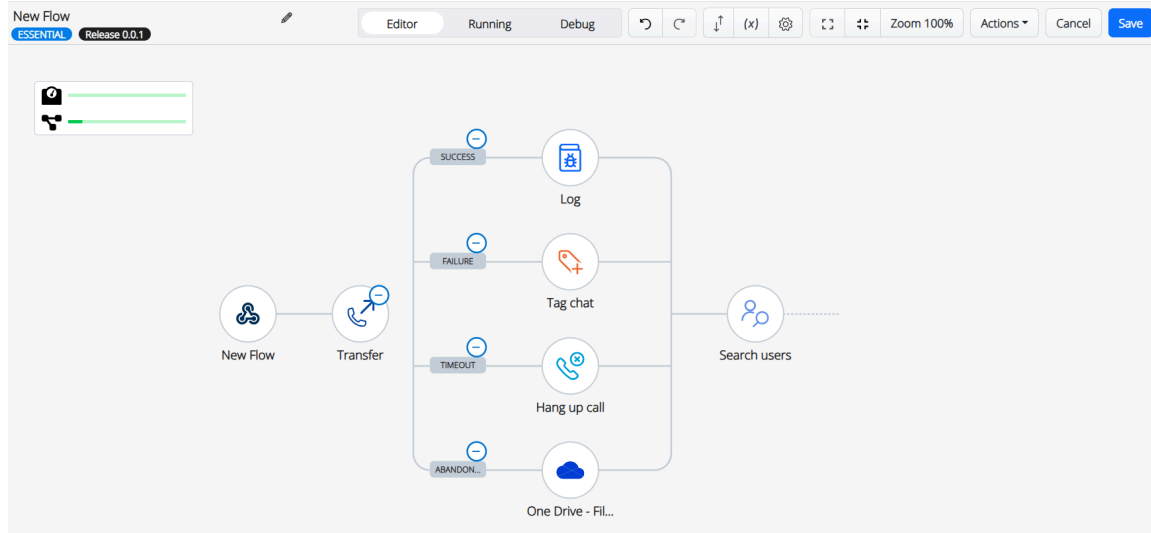


Figure 16: Completed flow

14. Click **Save**.

The new flow is created with the provided details.

5.2.2 Edit a flow

This topic provides step-by-step instructions for editing an existing workflow.

1. On the **Workflow Studio** home page, click **Flows**.

The **Flows** page is displayed.

2. On the **Flows** page, click the **Edit** (🔧) icon next to the existing workflow you want to modify.

The **Edit Flow** page is displayed with the existing flow.

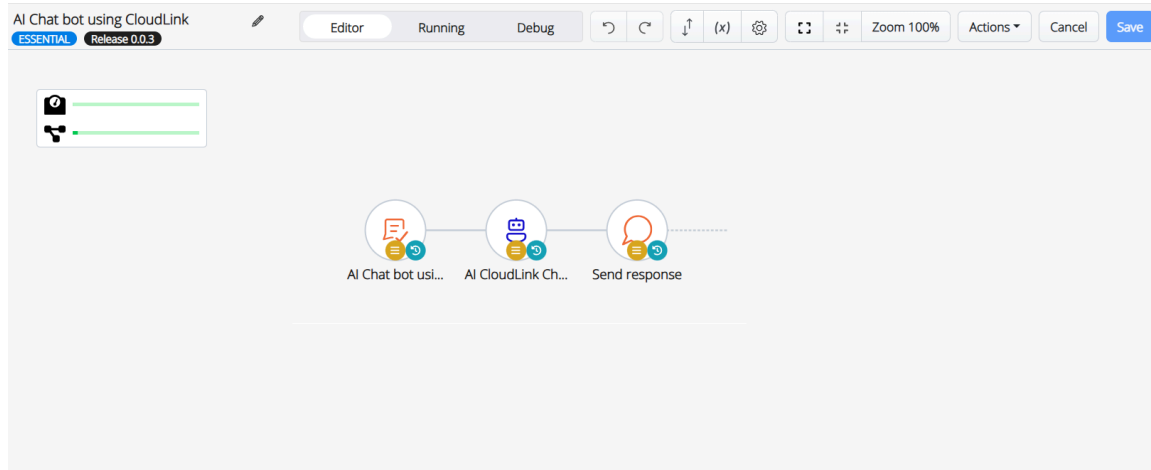


Figure 17: Edit Flow

3. Modify the workflow as required. Refer to the **Create a Flow** section for detailed instructions.

4. Click **Save**.

The flow is modified.

5.2.3 Import a flow

This topic provides the instructions for importing a workflow. The definition is supported in JSON format.

1. On the **Workflow Studio** home page, click **Flows**.

The **Flows** page is displayed.

2. On the **Flows** page, click **Import flow**.

The **Import Flow** popup screen is displayed.

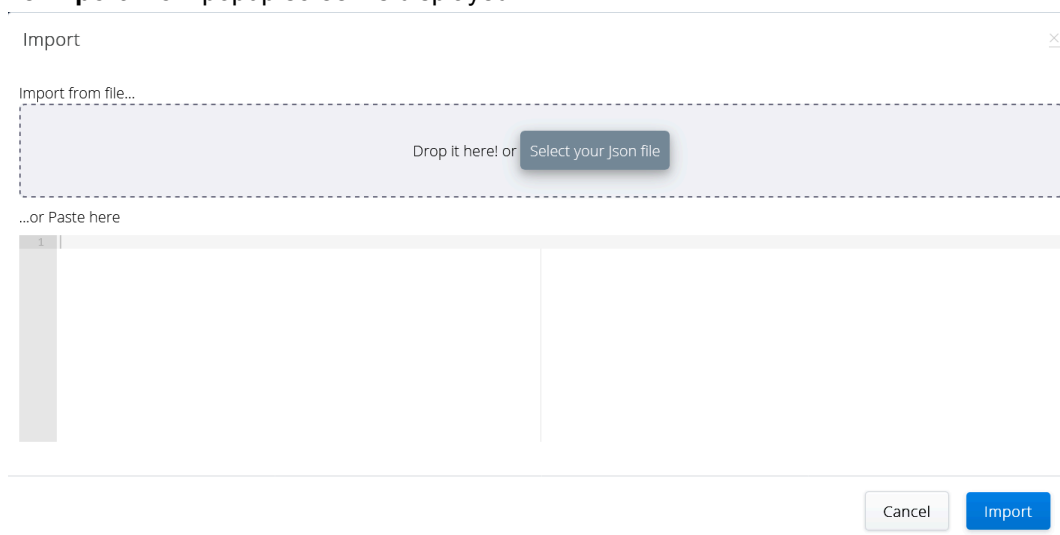


Figure 18: Import Flow

3. In the **Import** tab, perform one of the following actions to import the definition of the flow:

- a. Drag and drop the JSON file containing the flow definition in the **Import from file** field.
- b. Click **Select your json file** to select the JSON file from your local folder.
- c. Copy and paste the flow definition under the **Paste here** section.

4. Click **Import**.

The selected workflow is imported to the Workflow Canvas.

5.2.4 Clone a flow

This topic provides step-by-step instructions for cloning an existing workflow.

1. On the **Workflow Studio** home page, click **Flows**.

The **Flows** page is displayed.

2. On the **Flows** page, click the **Clone** (📄) icon next to the workflow you want to clone.
The **Workflow Canvas** page is displayed with the cloned flow.
3. Modify the workflow as required and click **Save**.

5.2.5 Delete a single or multiple flows

This topic provides step-by-step instructions for deleting the existing workflows individually or in bulk.

To delete a single workflow, perform the following steps:

1. On the **Workflow Studio** home page, click **Flows**.
The **Flows** page is displayed.
2. On the **Flows** page, click the **Delete** (🗑️) icon next to the workflow you want to delete.
The **Delete Selected Flow** popup screen is displayed.

Delete Selected Flow?

Deleting a workflow cannot be reversed.

Type delete in the box below to continue.

delete*

Cancel
Delete

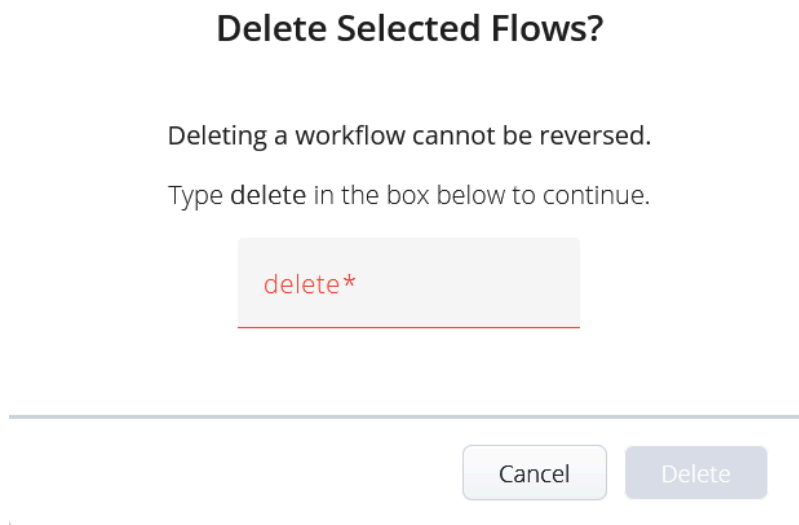
Figure 19: Delete Selected Flow

3. Enter the word “**delete**” in the text box to confirm the deletion.
4. Click **Delete**.

To delete multiple workflows, perform the following steps:

1. On the **Workflow Studio** home page, click **Flows**.
The **Flows** page is displayed.
2. On the **Flows** page, select the checkbox next to the workflows you want to delete.

3. Click the three-dots (⋮) icon on the top right of the page and select **Delete Selected**.
The **Delete Selected Flows** popup screen is displayed.



Delete Selected Flows?

Deleting a workflow cannot be reversed.
Type delete in the box below to continue.

delete*

Cancel Delete

Figure 20: Delete Selected Flows

4. Enter the word “**delete**” in the textbox to confirm the deletion.
5. Click **Delete**.

5.3 Library

The **Library** comprises a collection of available Actions, Triggers, Templates, and Integrations accessible to the account. You can easily browse and utilize predefined components to enhance and customize your workflows efficiently.

5.3.1 Actions

Actions are the fundamental building blocks that enable automation within workflows. Global actions are pre-configured actions with built-in functionalities. Local actions are custom-created actions to suit specific integration needs. Actions can range from simple tasks such as sending emails or updating database records to more complex operations like invoking external APIs or triggering notifications. You can orchestrate the business processes seamlessly by sequencing and configuring actions within workflows, automating repetitive tasks, and reducing manual intervention.

The Actions page displays a list of all actions available for the account, including all global actions and any local or composite actions created by other users.

You can access the **Actions** page by clicking **Library** on the left menu and selecting **Actions**.

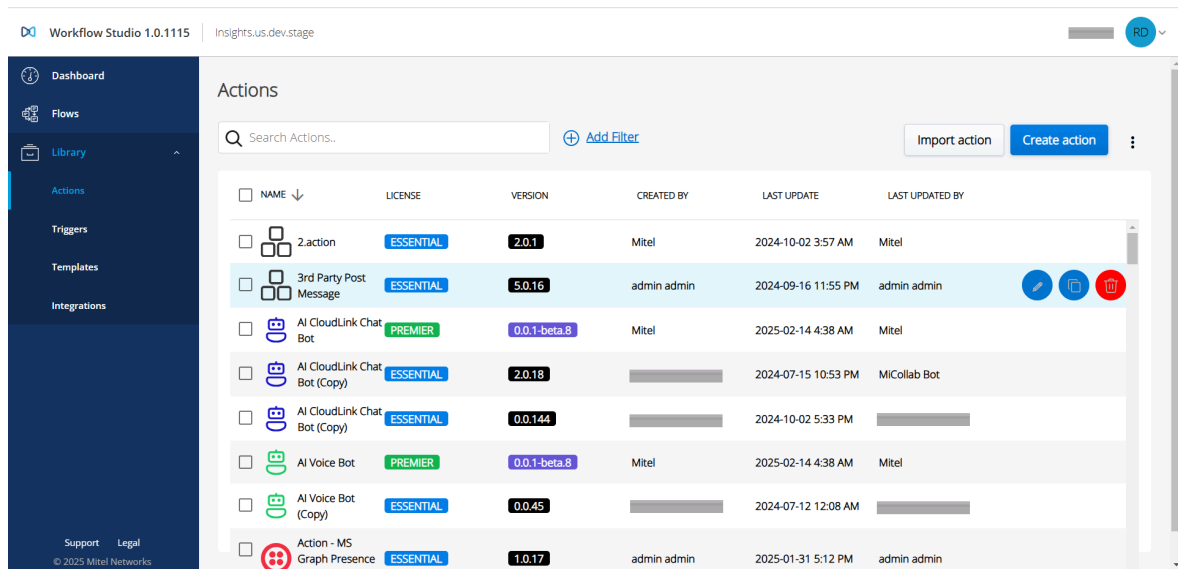


Figure 21: Actions

The following actions can be performed under the **Actions** section:

- **Create an action**
- **Edit an action**
- **Import an action**
- **Clone an action**
- **Delete a single or multiple action(s)**

5.3.1.1 Create an action

This topic provides step-by-step instructions for creating an action.

1. On the **Workflow Studio** home page, under **Library**, click **Actions**.

The **Actions** page is displayed.

2. On the **Actions** page, click **Create action**.

The **Create action** page is displayed.

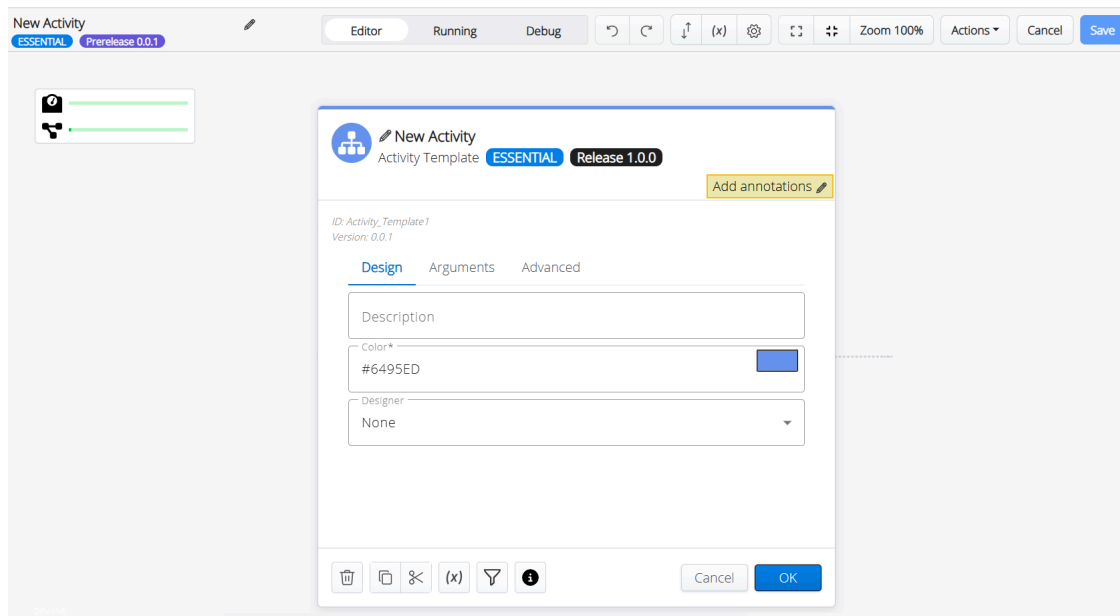



Figure 22: Create action

3. Enter the required parameters for the activity.
4. Click **OK**.

The **Activity** is added to the action.

5. Click  icon to add the next activity.
The **Search Activities** wheel is displayed.

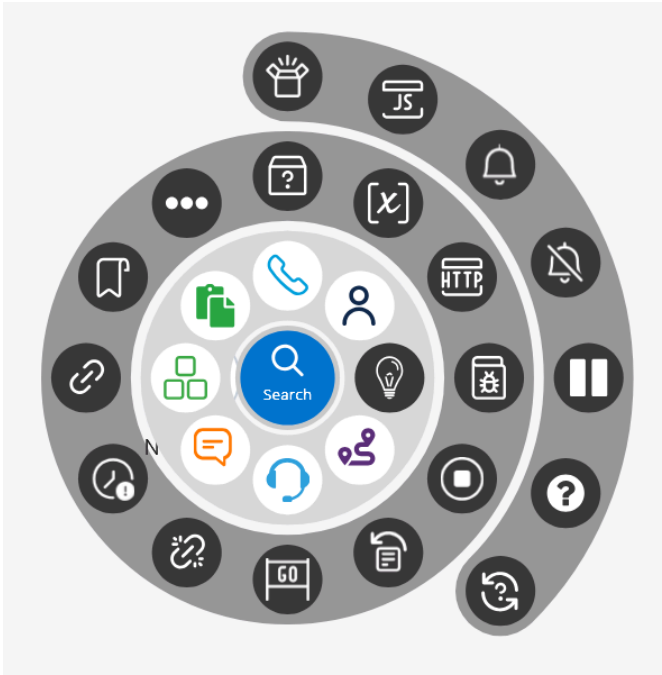


Figure 23: Search Activities

6. Search and select the required activity from the wheel.

The **Activity Parameters** screen is displayed.

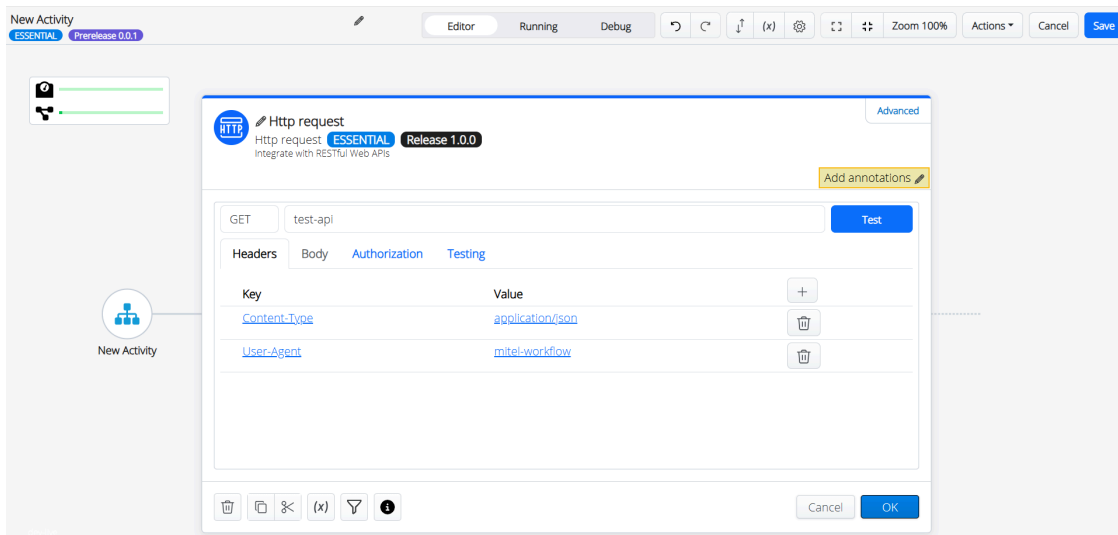


Figure 24: Activity Parameters

7. Provide the required details in the **Activity Parameters** screen.
8. Click **OK**.
9. Follow the above steps to add all the required activities for the action.

The **Completed action** screen is displayed with all the required activities.

10. Click **Save**.

The new action is created with the provided details.

5.3.1.2 Edit an action

This topic provides step-by-step instructions for editing an existing action. Global actions cannot be modified; they must be cloned to another local action for customization.

1. On the **Workflow Studio** home page, under **Library**, click **Actions**.

The **Actions** page is displayed.

2. On the **Actions** page, click the **Edit** (✎) icon next to the action you want to modify.

The **Edit Action** page is displayed with the existing action.

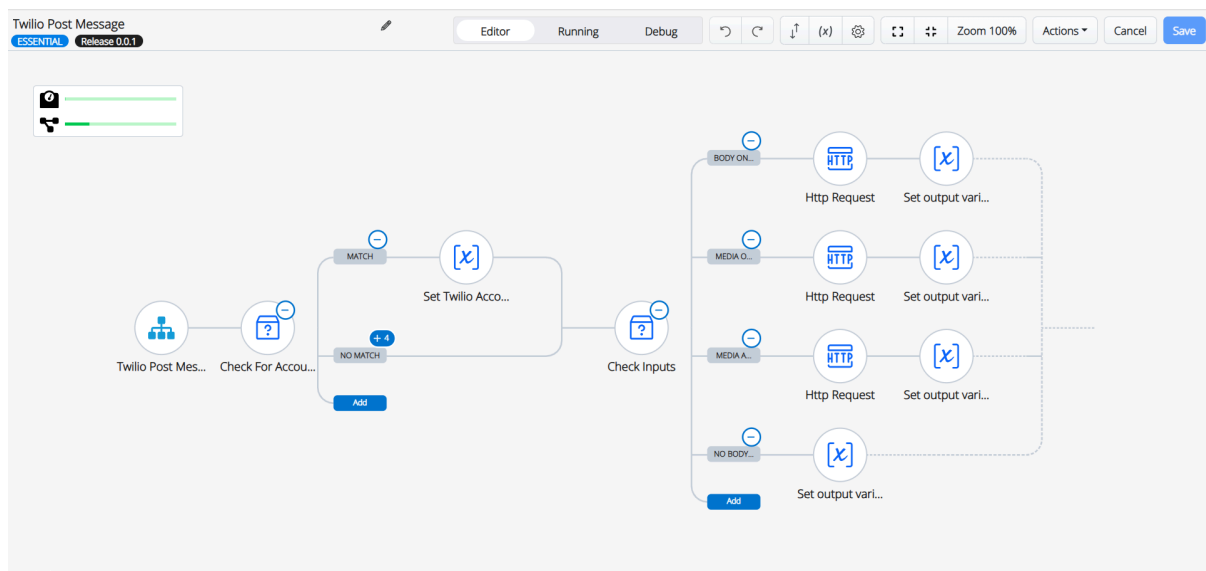


Figure 25: Edit Action

3. Modify the action as required. Refer to the **Create an action** section for detailed instructions.

4. Click **Save**.

5.3.1.3 Import an action

This topic provides step-by-step instructions for importing an action. The definition is supported in JSON format.

1. On the **Workflow Studio** home page, under **Library**, click **Actions**.

The **Actions** page is displayed.

2. On the **Actions** page, click **Import action**.

The **Import Action** popup screen is displayed.

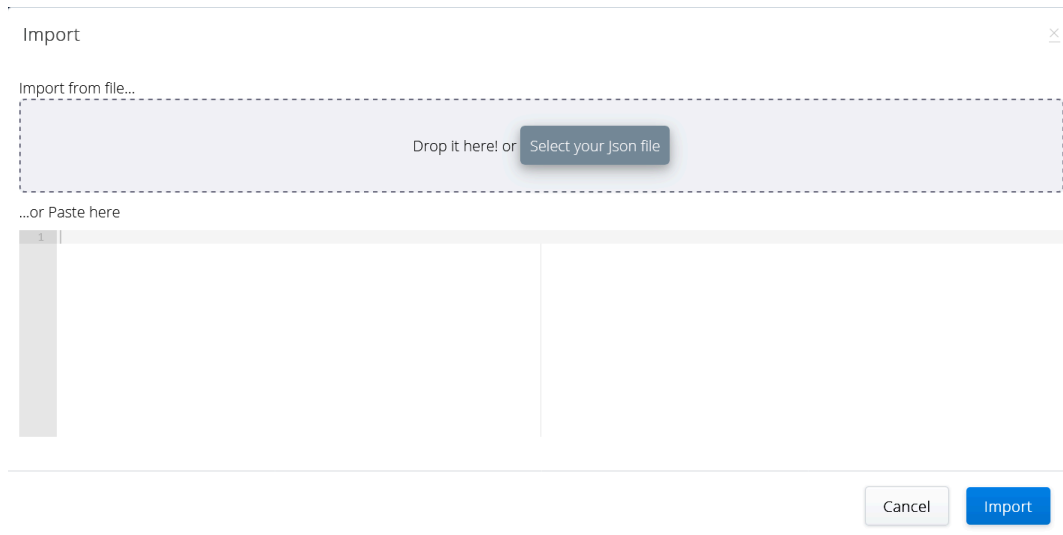


Figure 26: Import Action

3. In the **Import** tab, perform one of the following actions to import the definition of the action:
 - a. Drag and drop the JSON file containing the action definition in the **Import from file** field.
 - b. Click **Select your json file** to select the JSON file from your local folder.
 - c. Copy and paste the action definition under the **Paste here** section.
4. Click **Import**.

The selected action is imported to the canvas.

5.3.1.4 Clone an action

This topic provides step-by-step instructions for cloning an existing action.

1. On the **Workflow Studio** home page, under **Library**, click **Actions**.
The **Actions** page is displayed.
2. On the **Actions** page, click the **Clone** (📄) icon next to the action you want to clone.
The **Workflow Canvas** page is displayed with the cloned action.
3. Modify the action as required and click **Save**.

5.3.1.5 Delete a single or multiple actions

This topic provides step-by-step instructions for deleting the existing actions individually or in bulk.

**Note:**

Global actions cannot be deleted.

To delete a single action, perform the following steps:

1. On the **Workflow Studio** home page, under **Library**, click **Actions**.

The **Actions** page is displayed.

2. On the **Actions** page, click the **Delete** (🗑️) icon next to the action you want to delete.

The **Delete Selected Action** popup screen is displayed.

Delete Selected Action?

Deleting an action cannot be reversed.

Type delete in the box below to continue.

delete*

Cancel

Delete

Figure 27: Delete Selected Action

3. Enter the word “**delete**” in lowercase in the text box to confirm the deletion.
4. Click **Delete**.

To delete multiple actions, perform the following steps:

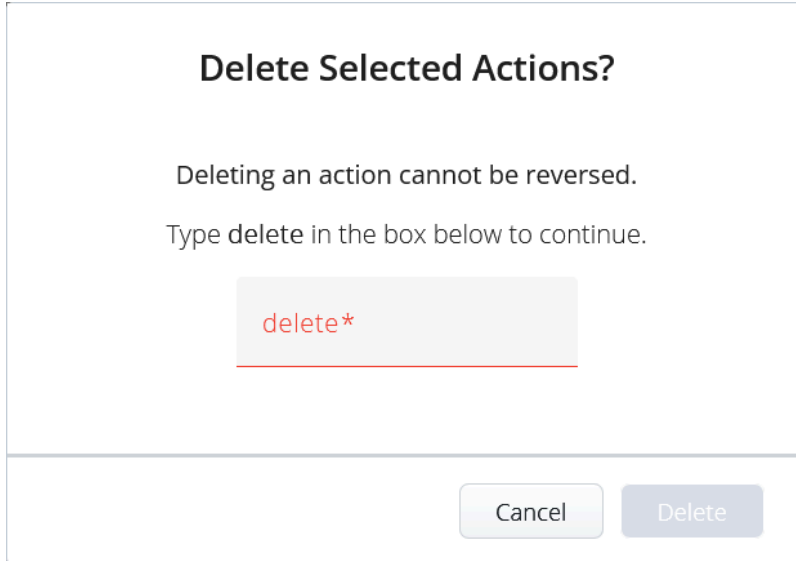
1. On the **Workflow Studio** home page, under **Library**, click **Actions**.

The **Actions** page is displayed.

2. On the **Actions** page, select the checkbox next to the actions you want to delete.

3. Click the three-dots (⋮) icon on the top right of the page and select **Delete Selected**.

The **Delete Selected Actions** popup screen is displayed.



Delete Selected Actions?

Deleting an action cannot be reversed.
Type delete in the box below to continue.

delete*

Cancel Delete

Figure 28: Delete Selected Actions

4. Enter the word “**delete**” in the textbox to confirm the deletion.
5. Click **Delete**.

5.3.2 Triggers

Triggers are configured to monitor designated conditions within the system and automatically launch associated workflows when specific criteria is met. Workflow Studio uses the triggers to respond dynamically to changes or inputs, ensuring timely and efficient automation of business processes.

The **Triggers** screen displays a list of all triggers available for the account, including all global triggers and any local triggers.

You can access the **Triggers** page by clicking **Library** on the left menu and selecting **Triggers**.

The **Triggers** page is displayed.

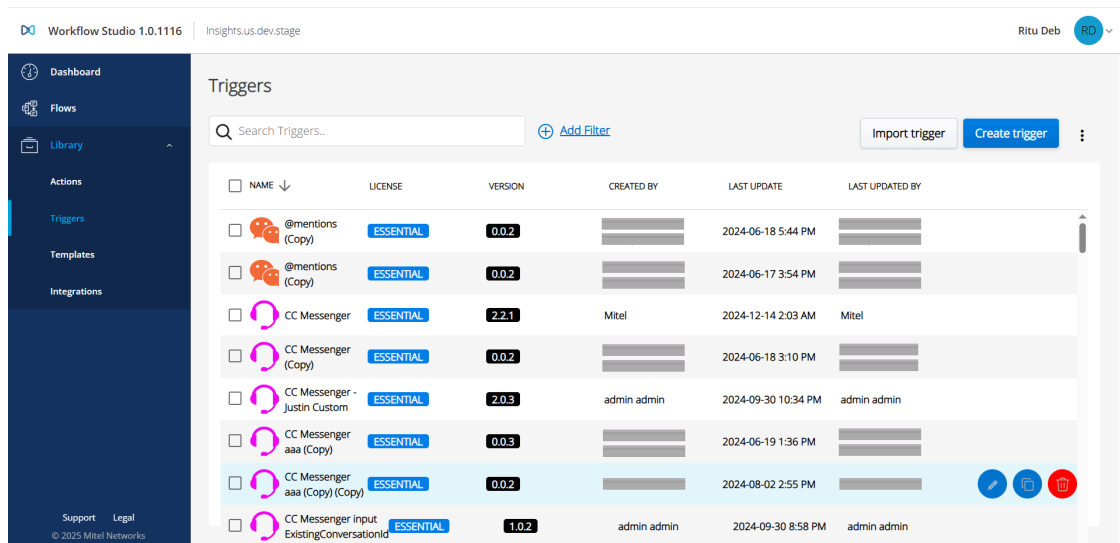


Figure 29: Triggers

The following actions can be performed under the **Triggers** section:

- **Create a trigger**
- **Edit a trigger**
- **Import a trigger**
- **Clone a trigger**
- **Delete a single or multiple trigger(s)**

5.3.2.1 Create a trigger

This topic provides step-by-step instructions for creating a new trigger.

1. On the **Workflow Studio** home page, under **Library**, click **Triggers**.

The **Triggers** page is displayed.

2. On the **Triggers** page, click **Create trigger**.

The **Create trigger** page is displayed.

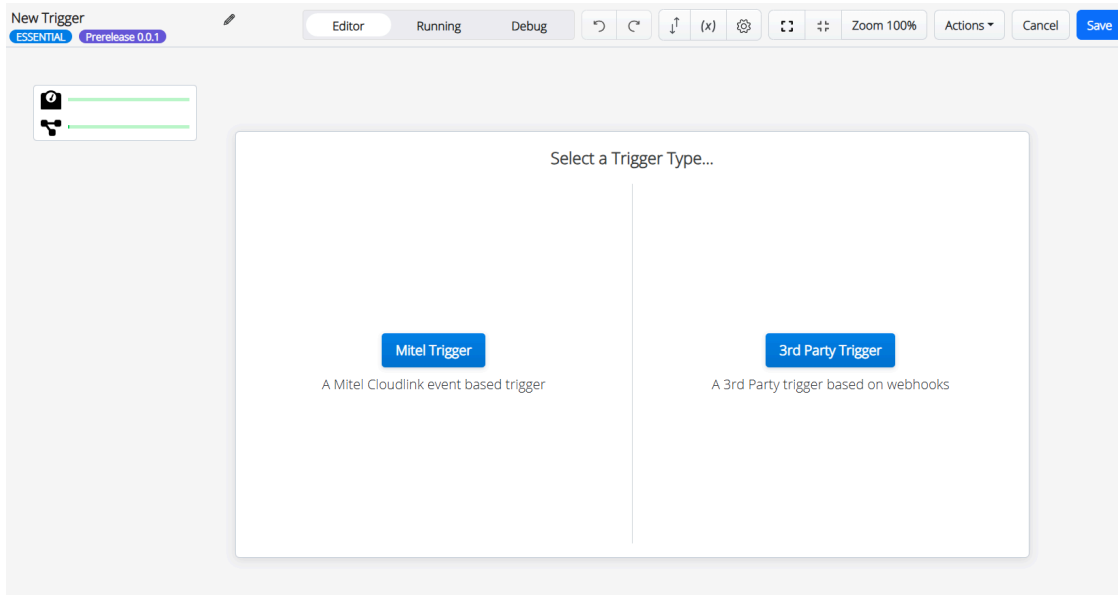


Figure 30: Create trigger

3. Select the type of trigger from the following options:

- a. Mitel Trigger – will trigger based on a Mitel CloudLink event.
- b. 3rd Party Trigger – will trigger based on an event from a third-party application.

The **Trigger Parameters** screen is displayed based on the selected trigger type.

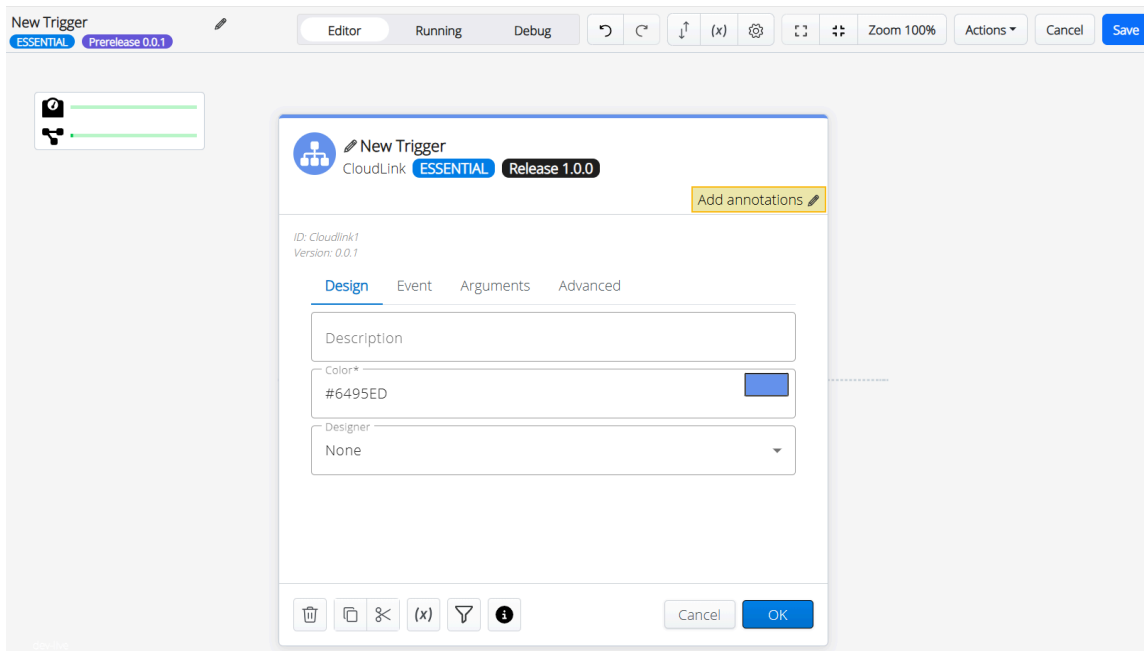



Figure 31: Trigger Parameters

4. Enter the required parameters in the **Trigger Parameters** screen.
5. Click **OK**.

The **Trigger** is added to the **Workflow Canvas**.

6. Click  icon to add the next activity to the trigger.

The **Search Activities** wheel is displayed.

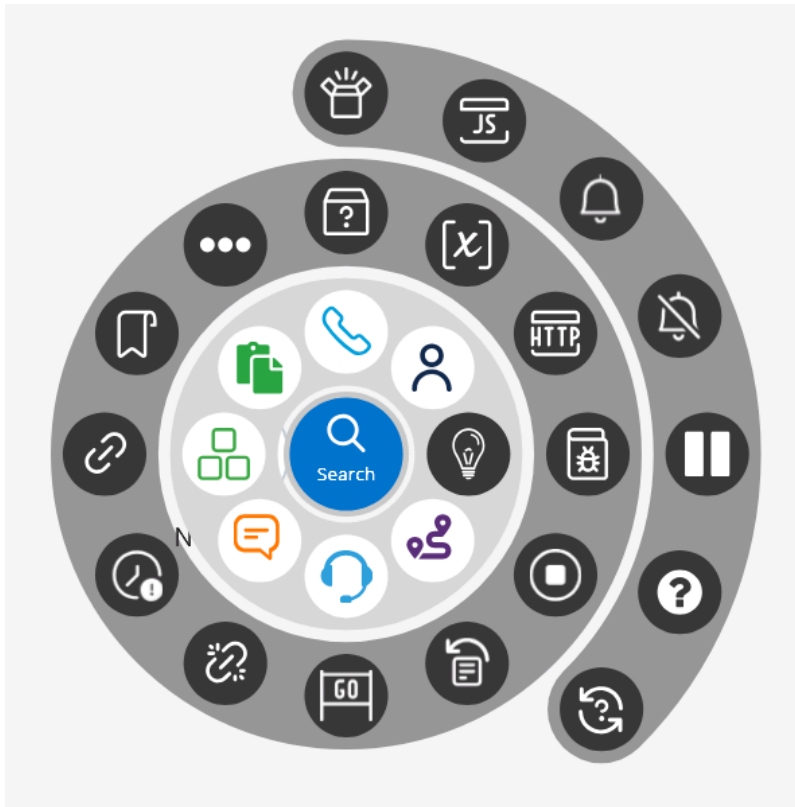


Figure 32: Search Activities

7. Search and select the required activity from the wheel.

The **Activity Parameters** screen is displayed.

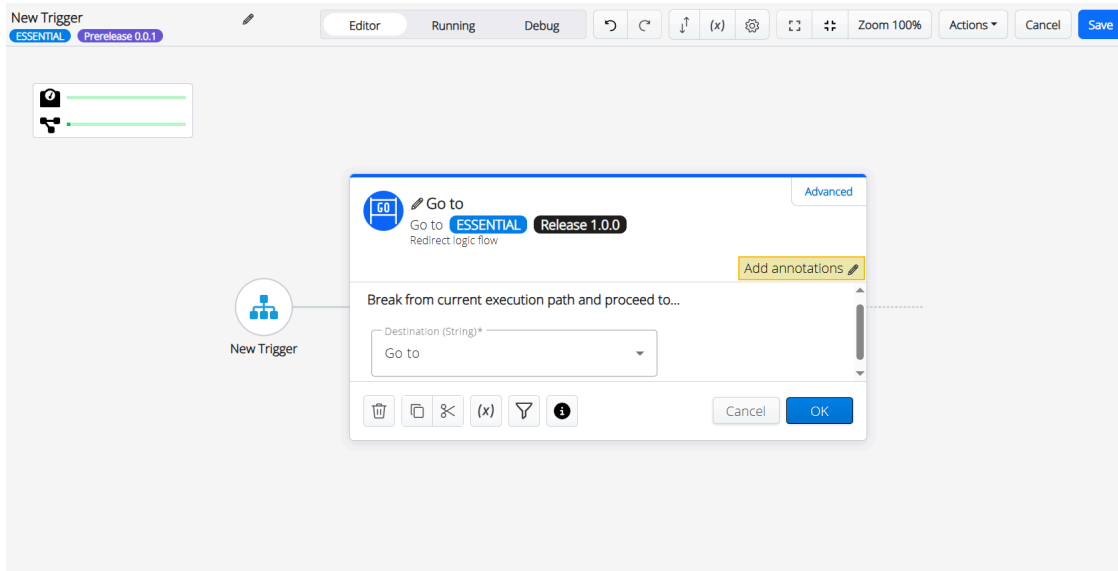


Figure 33: Add Activity Parameters

8. Provide the required details in the **Activity Parameters** screen.
9. Click **OK**.
10. Follow the above steps to add all the required activities for the trigger.

The **Completed Trigger** screen is displayed with all the required activities.

11. Click **Save**.

The new trigger is created with the provided details.

5.3.2.2 Edit a trigger

This topic provides step-by-step instructions for editing the existing trigger. Global triggers cannot be modified; they must be cloned to another local trigger for customization.

1. On the **Workflow Studio** home page, under **Library**, click **Triggers**.

The **Triggers** page is displayed.

2. On the **Triggers** page, click the **Edit** (✎) icon next to the trigger you want to modify.

The **Edit Trigger** page is displayed with the existing action.

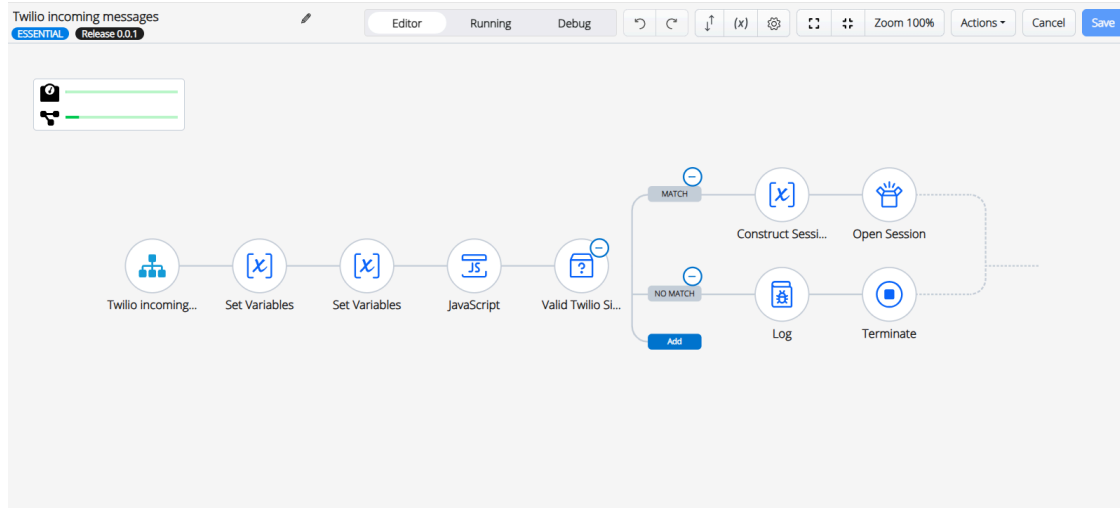


Figure 34: Edit Trigger

3. Modify the trigger as required. Refer to the **Create a trigger** section for detailed instructions.
4. Click **Save**.

5.3.2.3 Import a trigger

This topic provides step-by-step instructions for importing the trigger. The flow definition is supported in JSON format.

1. On the **Workflow Studio** home page, under **Library**, click **Triggers**.

The **Triggers** page is displayed.

2. On the **Triggers** page, click **Import trigger**.

The **Import Trigger** popup screen is displayed.

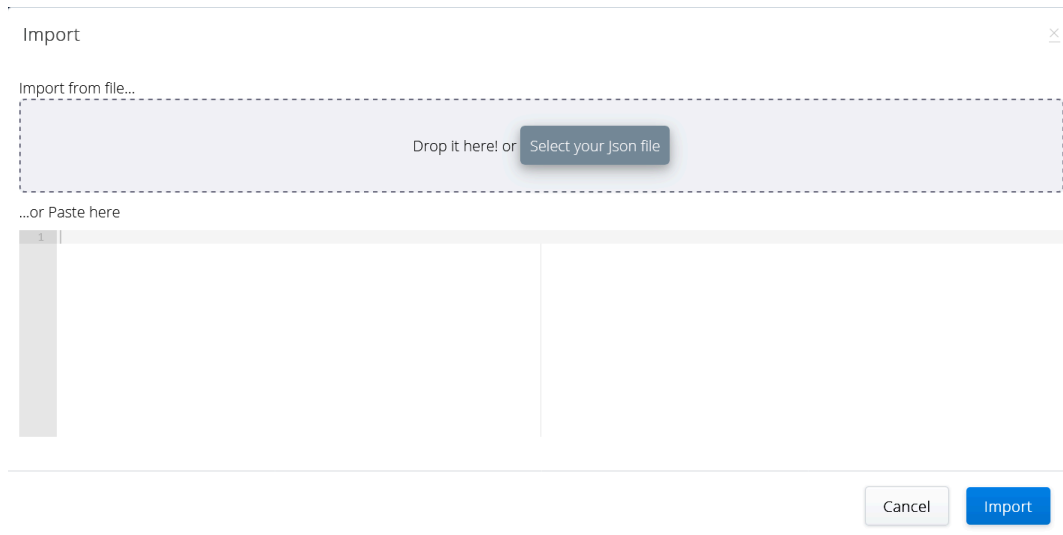



Figure 35: Import Trigger

3. In the **Import** tab, perform one of the following actions to import the definition of the trigger:
 - a. Drag and drop the JSON file containing the trigger definition in the **Import from file** field.
 - b. Click **Select your json file** to select the JSON file from your local folder.
 - c. Copy and paste the trigger definition under the **Paste here** section.
4. Click **Import**.

The selected trigger is imported to the canvas.

5.3.2.4 Clone a trigger

This topic provides step-by-step instructions for cloning an existing trigger.

1. On the **Workflow Studio** home page, under **Library**, click **Triggers**.
The **Triggers** page is displayed.
2. On the **Triggers** page, click the **Clone** () icon next to the trigger you want to clone.
The **Workflow Canvas** page is displayed with the cloned trigger.
3. Modify the trigger as required and click **Save**.

5.3.2.5 Delete a single or multiple triggers

This topic provides step-by-step instructions for deleting the existing triggers individually or in bulk.

Note:
Global triggers cannot be deleted.

To delete a single trigger, perform the following steps:

1. In the **Workflow Studio** home page, under **Library**, click **Triggers**.

The **Triggers** page is displayed.

2. In the **Triggers** page, click the **Delete** (🗑️) icon next to the trigger you want to delete.

The **Delete Selected Trigger** popup screen is displayed.

Delete Selected Trigger?

Deleting a trigger cannot be reversed.

Type delete in the box below to continue.

delete*

Cancel

Delete

Figure 36: Delete Selected Trigger

3. Enter the word “**delete**” in the text box to confirm the deletion.
4. Click **Delete**.

To delete multiple triggers, perform the following steps:

1. On the **Triggers** page, select the checkbox next to the triggers you want to delete.

2. Click the three-dot (⋮) icon on the top right of the page and select **Delete Selected**.
The **Delete Selected Triggers** popup screen is displayed.

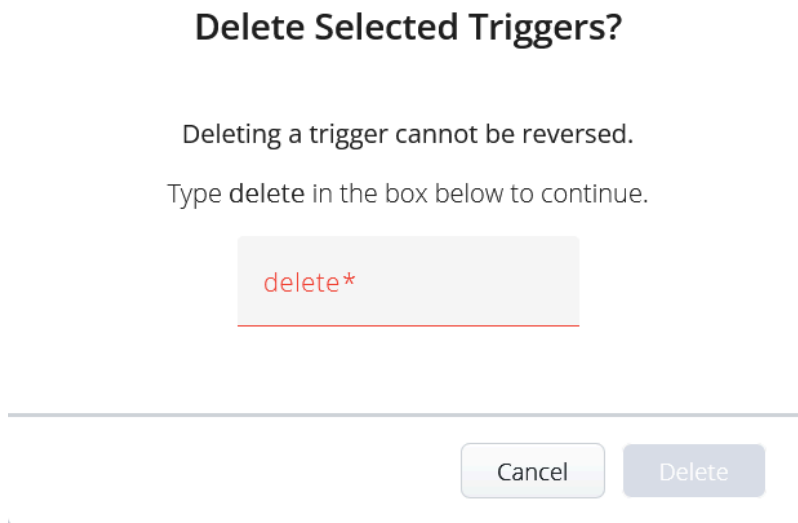


Figure 37: Delete Selected Triggers

3. Enter the word “**delete**” in the textbox to confirm the deletion.
4. Click **Delete**.

5.3.3 Templates

The **Templates** section displays a list of workflow templates. Templates cannot be created or edited but can be cloned to create a new flow. They provide examples for integrating flows with existing solutions, such as social media applications.

You can access the **Templates** page by clicking **Library** on the left menu and selecting **Templates**.

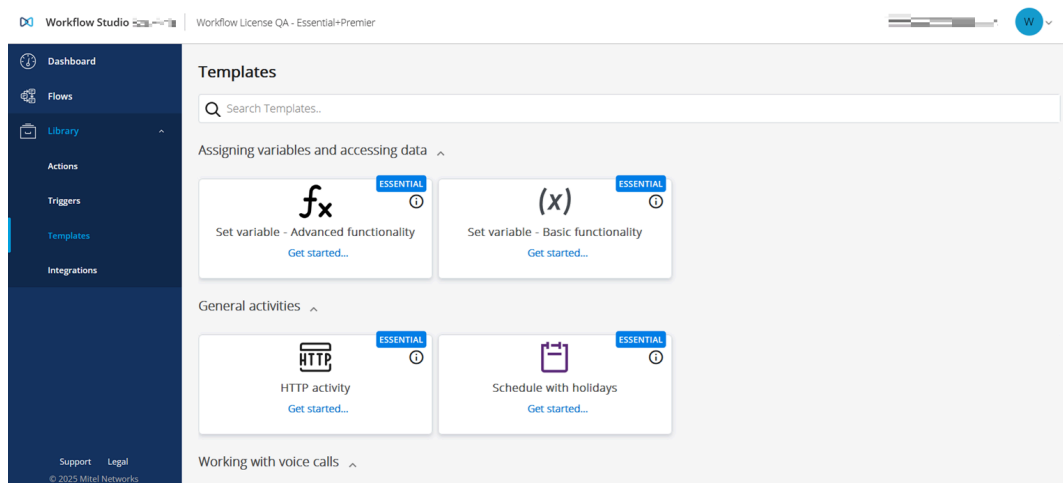


Figure 38: Templates

Click the **Get started** link to open the predefined template in the Workflow canvas and modify it based on your requirements.

5.3.4 Integrations

The **Integrations** section contains the list of integrated products. You can add the integrations through **Account Console** via Partner. Connections to existing integrations can be added and managed through **Workflow Studio**.

Refer to [Integrations](#) to view the examples for various integrations.

You can access the **Integrations Hub** page by clicking **Library** on the left menu and selecting **Integrations**.

The **Integrations Hub** page is displayed.

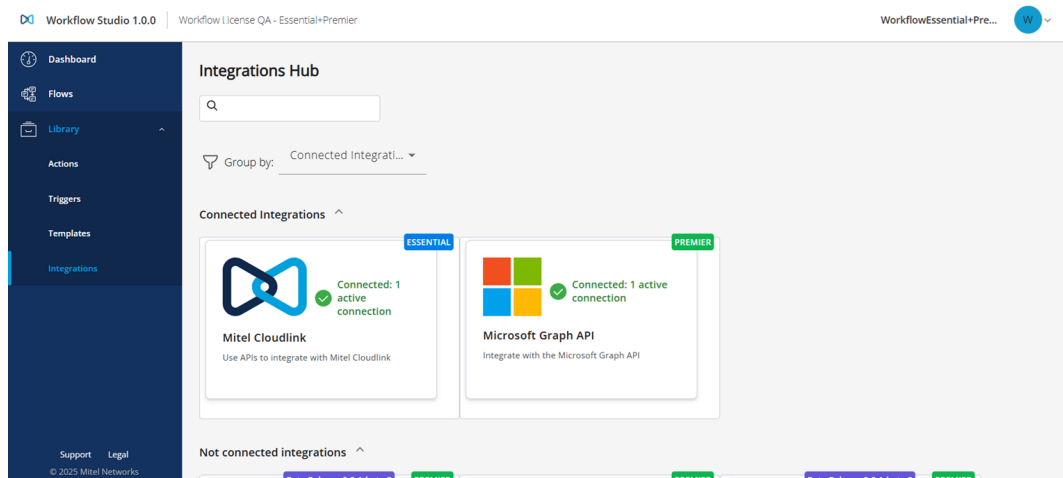


Figure 39: Integrations Hub

The following actions can be performed under the **Integrations Hub** section.

- **Add new connection**
- **Configure an existing connection**
- **Delete a connection**

5.3.4.1 Add new connection

This topic provides step-by-step instructions on how to add new connections for connected and not-connected products.

To add a new connection for a product that doesn't have any existing connection, perform the following steps:

1. On the **Workflow Studio** home page, under **Library**, click **Integrations**.

The **Integrations Hub** page is displayed.

2. On the **Integrations Hub** page, under the **Not connected integrations** section, click on the **See details** button next to the product you want to integrate.

The **Add new connection** page is displayed.

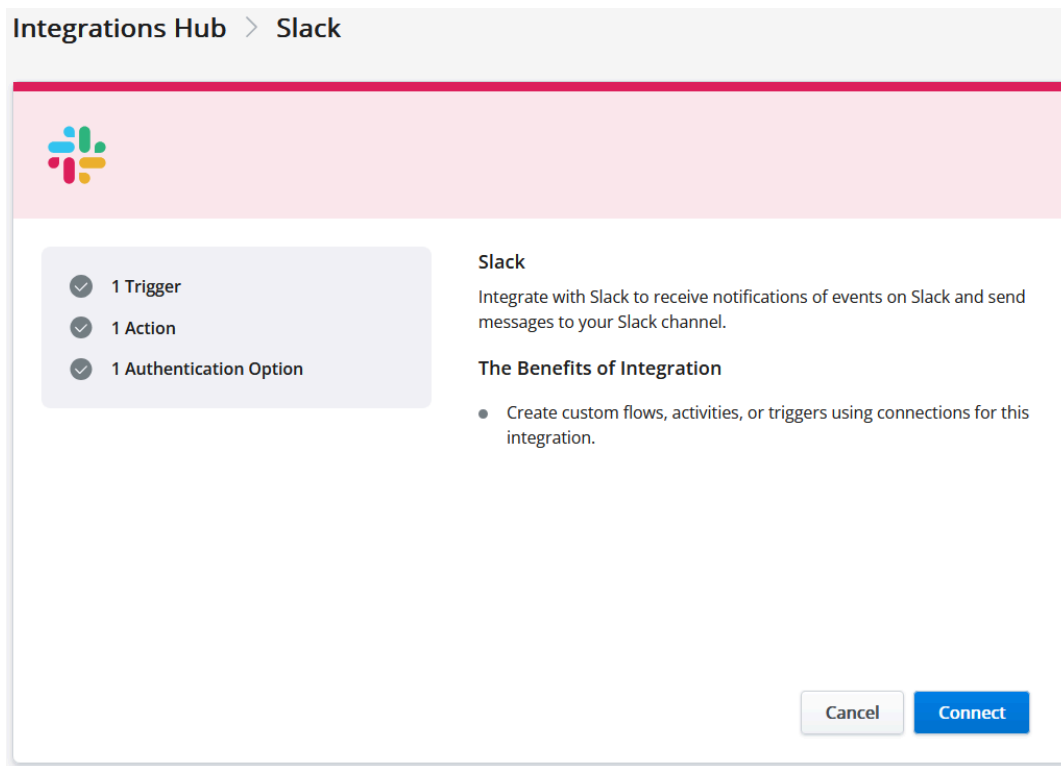


Figure 40: Add new connection

3. Click **Connect**.

The **Add new connection – Getting Started** page is displayed.

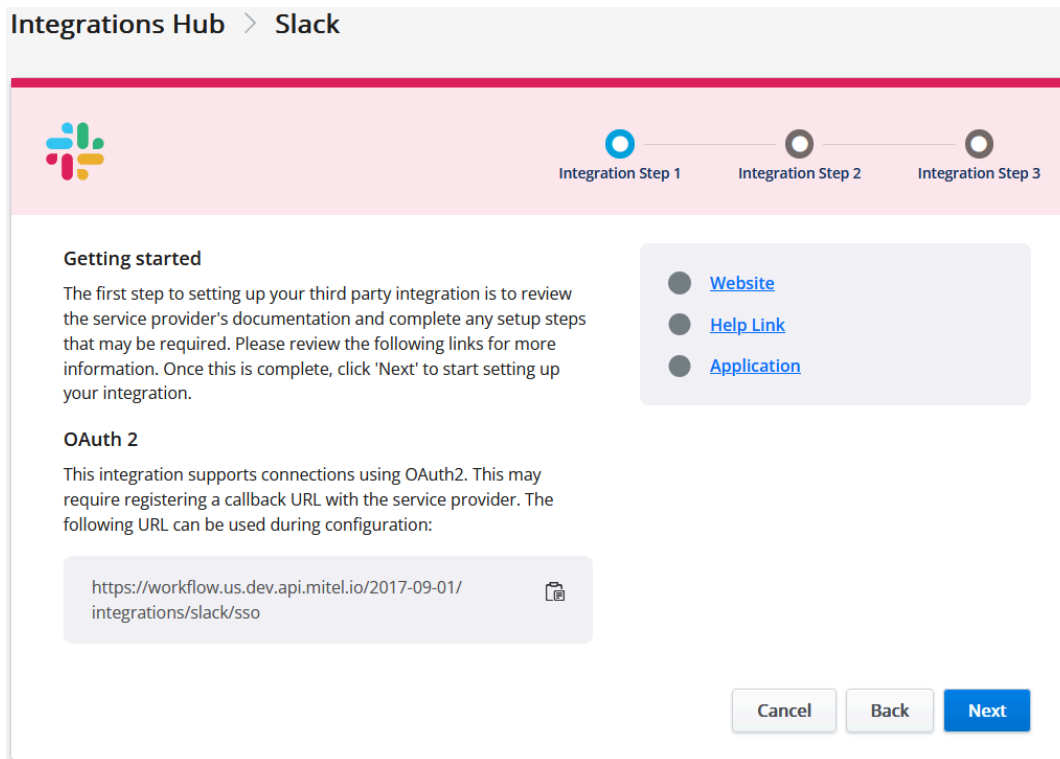


Figure 41: Add new connection – Getting Started

4. Click Next.

The **Add a connection to get started** page is displayed.

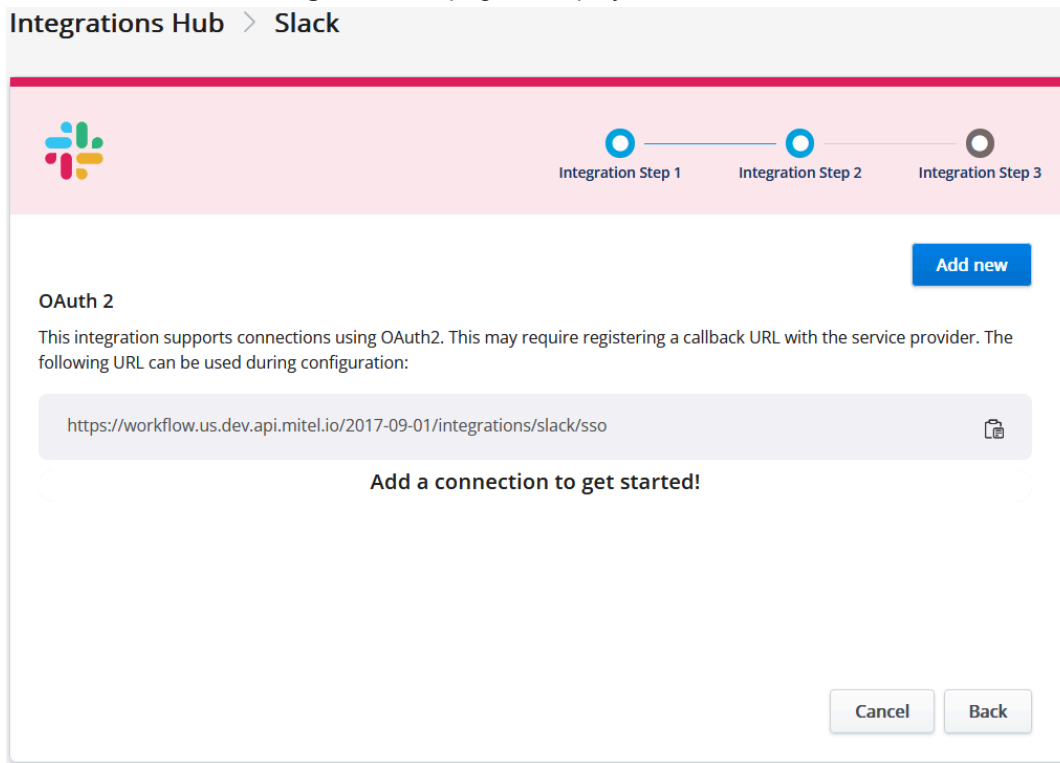


Figure 42: Add a connection to get started

5. Click **Next**.

The **Add new connection – Complete Setup** page is displayed.

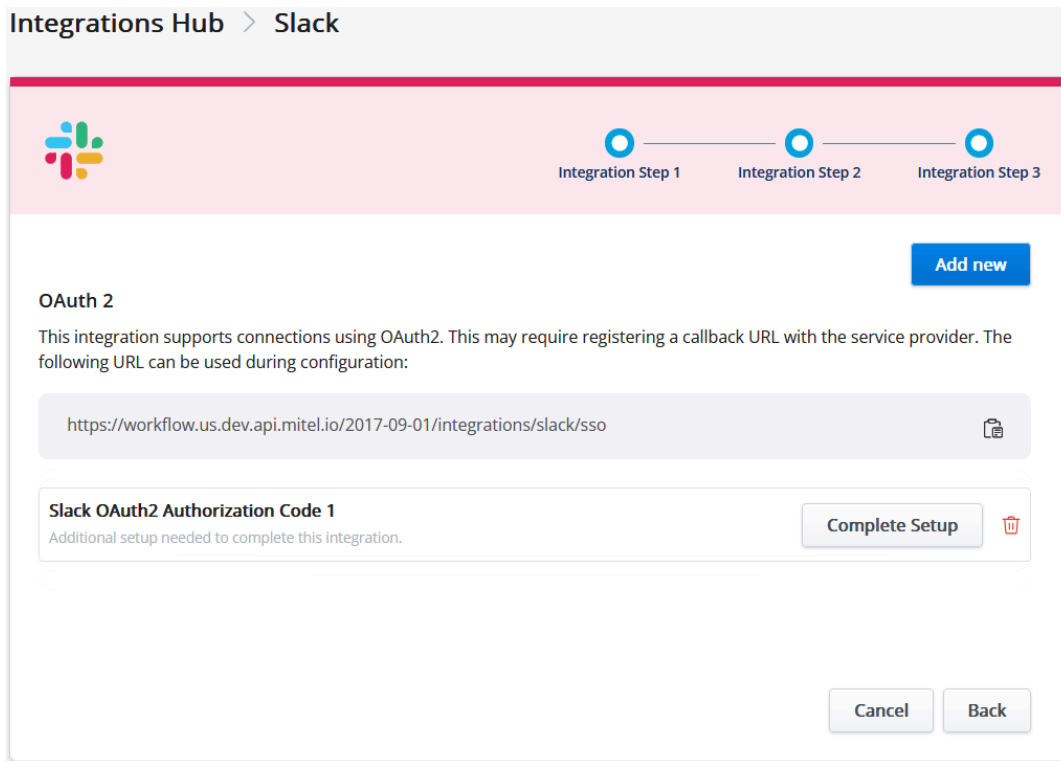


Figure 43: Add new connection – Complete Setup

6. Click **Complete Setup**.

The **Configure this connection** popup screen is displayed.

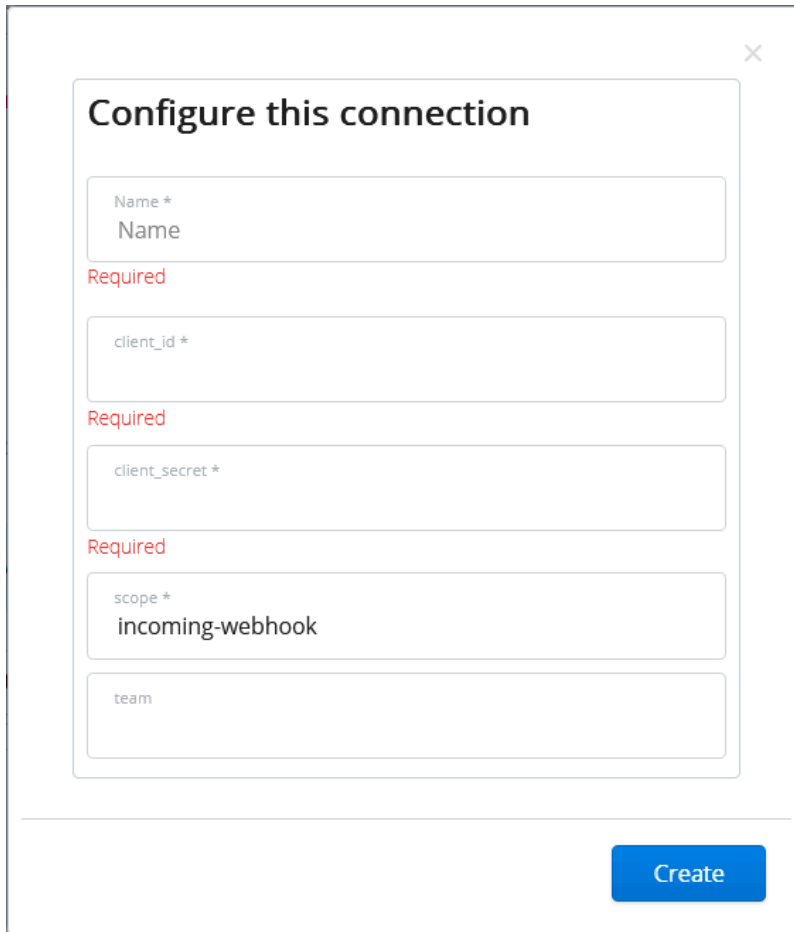


Figure 44: Configure this connection


7. On the **Configure this connection** screen, enter the required details in the respective fields.

8. Click **Create**.

The Success message for third-party integration is displayed.

To add a new connection for an integrated product, perform the following steps:

1.

On the **Integrations Hub** page, under the **Connected Integrations** section, click  icon next to the integrated product you want to configure.

2. Click **Add New**.

The **Add connection** popup screen is displayed when the integration has multiple connection types.

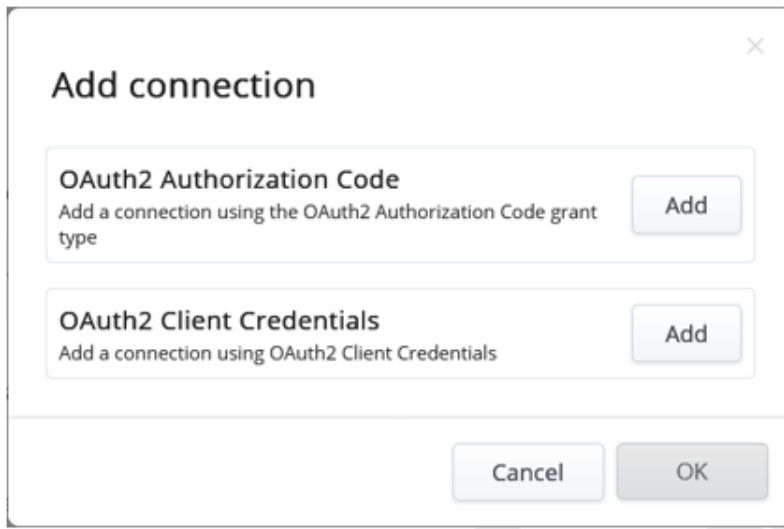


Figure 45: Add connection

3. Click **Add** next to the connection type you want to use for the integration.



Note:

The connection type will vary depending on the Integrations. Refer to the [Integrations](#) chapter to know the applicable connection type for each integration.

4. Click **OK**.

5. Click **Complete Setup**.

The **Configure this connection** popup screen is displayed.

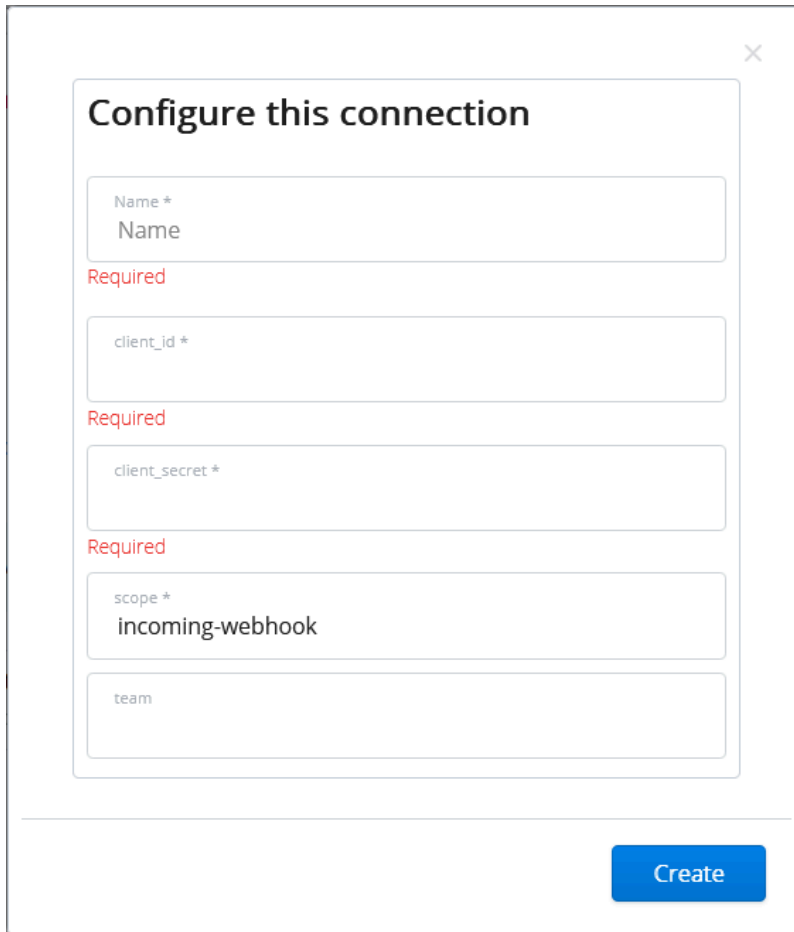


Figure 46: Configure this connection

Note:

The fields on this screen differ based on the added integration.

6. On the **Configure this connection** screen, enter the required details in the respective fields.

7. Click **Create**.

Upon successful connection, the success message for third-party integration is displayed.

5.3.4.2 Configure an existing connection


This topic provides step-by-step instructions on modifying an existing connection of the integrated product.

To modify an existing connection, perform the following steps:

1. On the **Workflow Studio** home page, under **Library**, click **Integrations**.

The **Integrations Hub** page is displayed.

- 2.

On the **Integrations Hub** page, under the **Connected Integrations** section, click  icon next to the integrated product you want to modify.

The **Integration Setup** page is displayed.

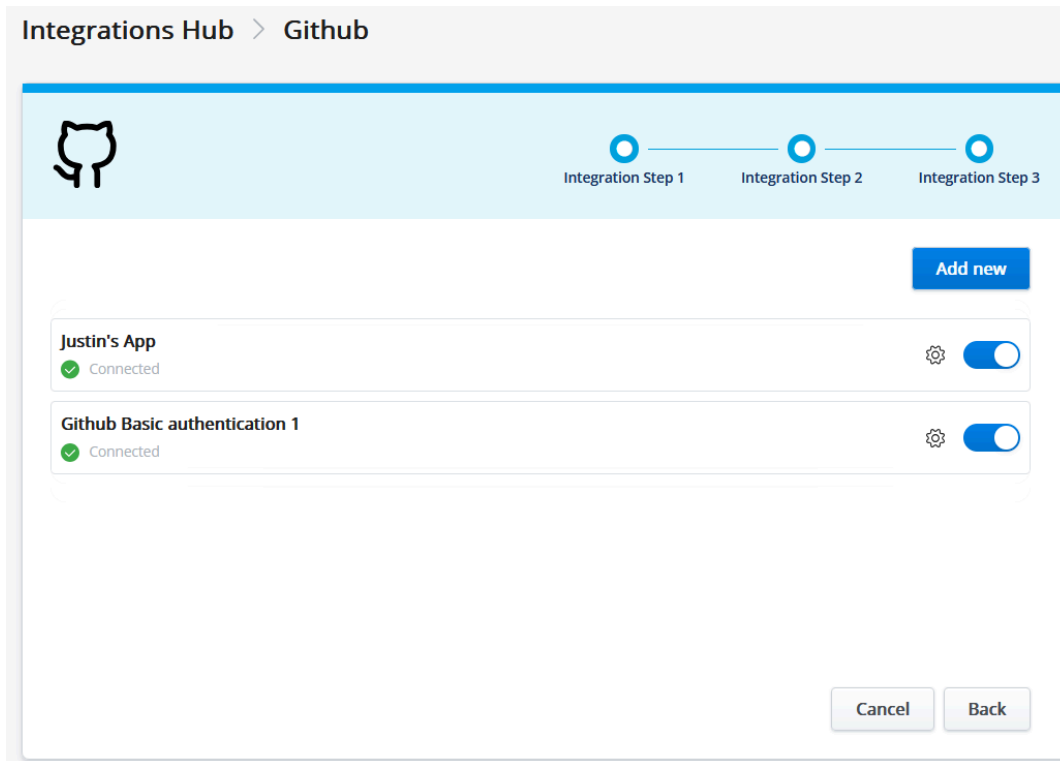



Figure 47: Integration Setup

- 3.

Click  icon to modify the connection details.

The **Configure this connection** popup screen is displayed with the existing details.

4. Modify the required details in the respective field.
5. Click **Save** to save the changes.

5.3.4.3 Delete a connection

This topic provides step-by-step instructions on how to delete a connection.

1. On the **Workflow Studio** home page, under **Library**, click **Integrations**.

The **Integrations Hub** page is displayed.

2. On the **Integrations Hub** page, under the **Connected integrations** section, click  icon next to the integrated product.

The **Integration Setup** page is displayed.

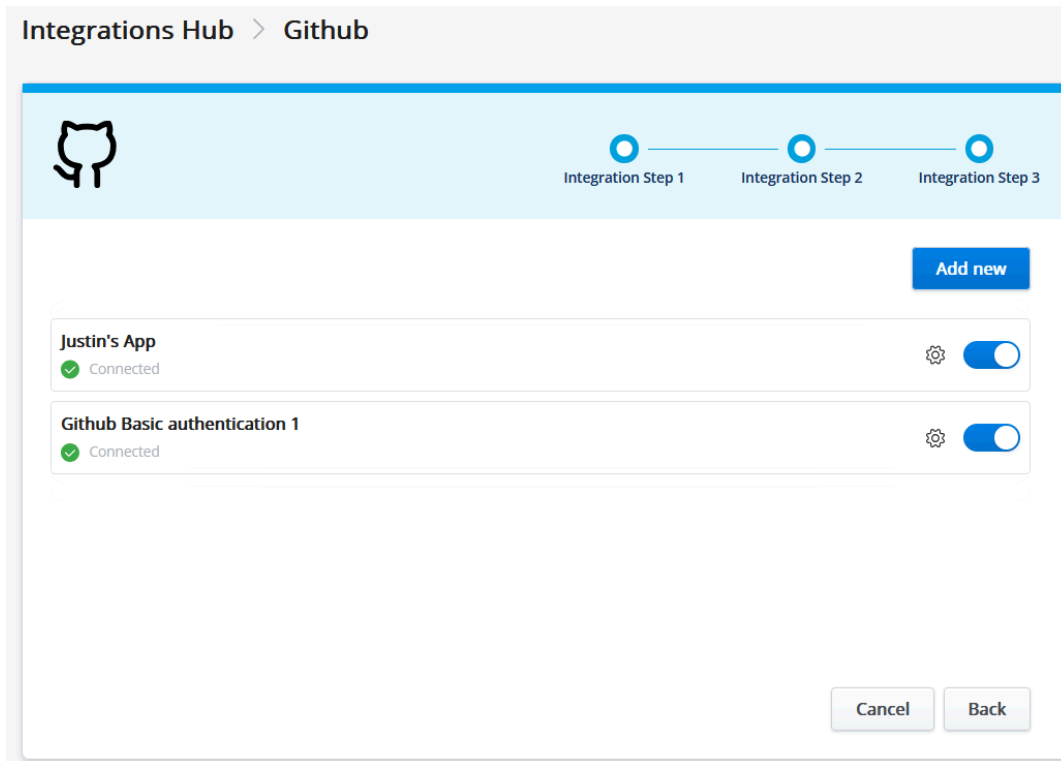
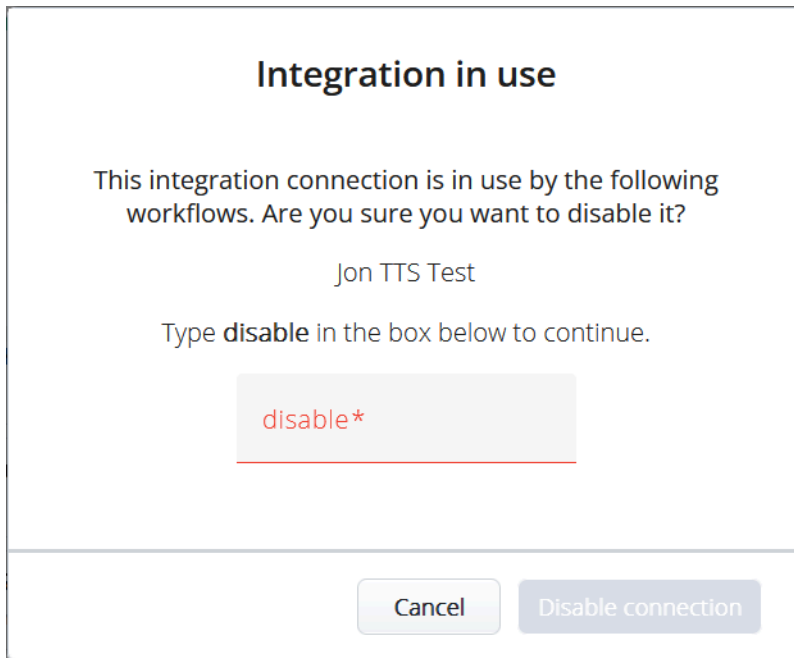


Figure 48: Integration Setup

3. Select the toggle to disable the connection you want to delete.

The system validates whether the connection is being used by an active workflow.

The **Integration in use** popup screen is displayed.



The image shows a modal dialog box titled "Integration in use". Inside the dialog, the text reads: "This integration connection is in use by the following workflows. Are you sure you want to disable it?". Below this text, the name of the workflow "Jon TTS Test" is listed. Further down, there is a prompt: "Type **disable** in the box below to continue." Below the prompt is a text input field containing the text "disable*" in red. At the bottom of the dialog, there are two buttons: "Cancel" and "Disable connection".

Figure 49: Integration in use

Note:

If the connection is not used in any active workflow, the **Integration in use** popup screen will not be displayed for the confirmation.

4. Enter the text "**disable**" in the text box.

5. Click **Disable connection** to confirm disabling the connection.

The **Integration Setup - Delete** page is displayed.

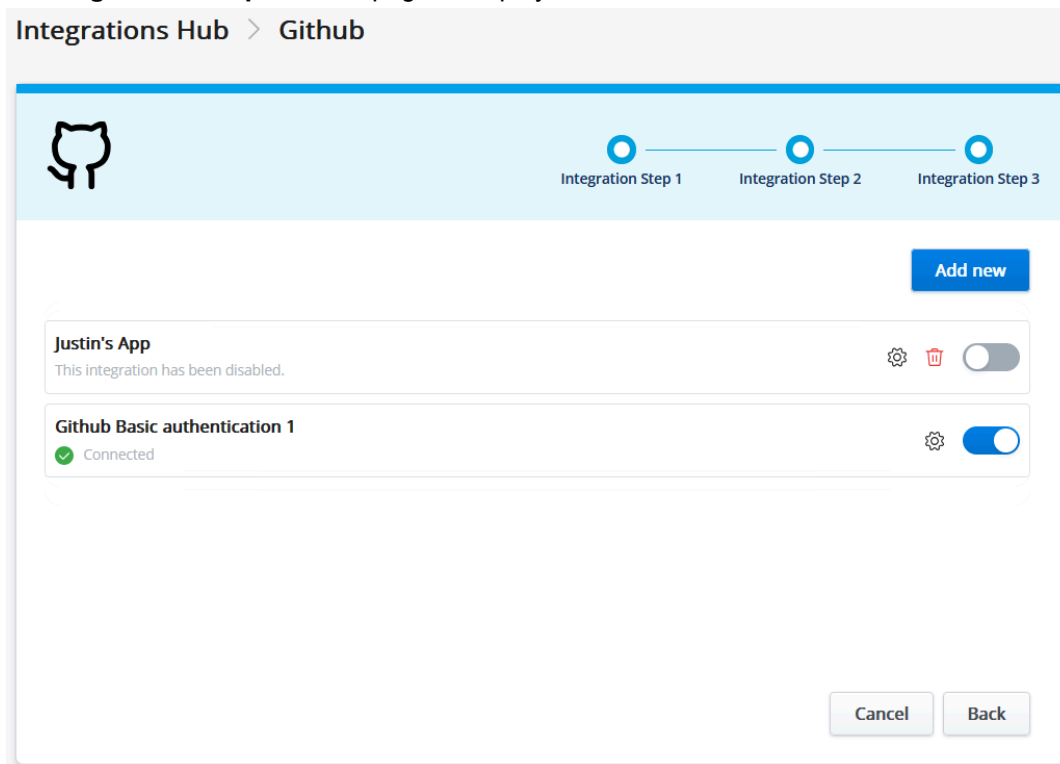


Figure 50: Integration Setup - Delete

6. Click  icon next to the toggle.

The **Delete connection** popup screen is displayed.

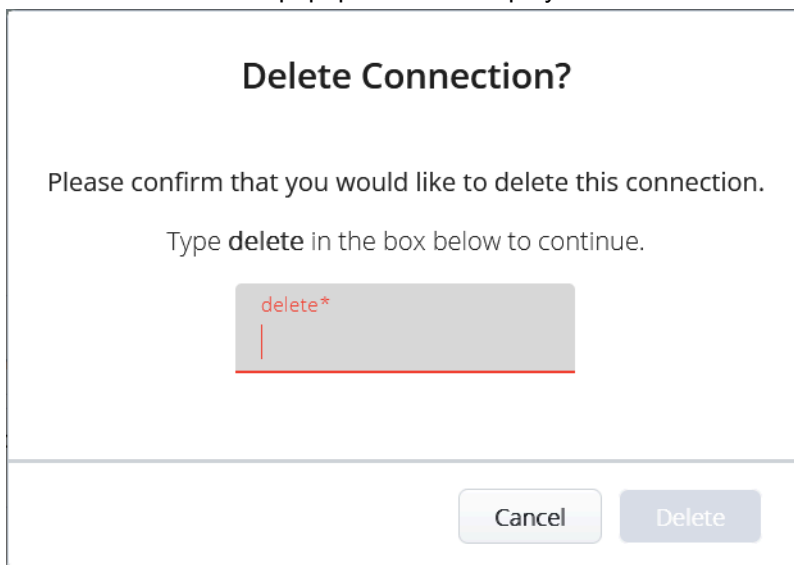


Figure 51: Delete Connection

7. Enter the text **"delete"** in the textbox to confirm the deletion.
8. Click **Delete**.

This chapter contains the following sections:

- [Views](#)
- [Basic Operations](#)
- [Activity Wheel](#)
- [Expressions](#)
- [Using Variables](#)

6.1 Views

As an administrator, you can create, modify, run, and debug the workflow using **Workflow Canvas**. You can also view the progress of the workflow.

The **Workflow Canvas** page consists of the following views:

- Editor
- Running
- Debug

6.1.1 Editor

The **Editor** view is the default view of the **Workflow Canvas**, which allows the administrator to create or modify the workflow based on the requirements. The **Progress Bars** are provided in the **Editor** view to view the size and number of nodes in the flow/action.

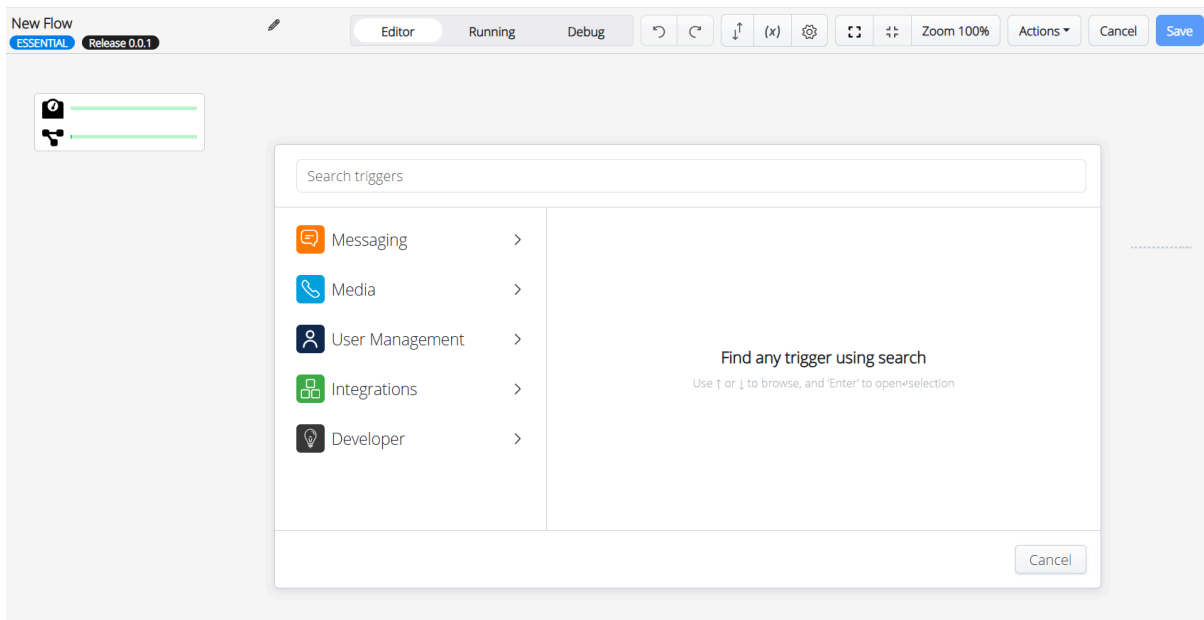


Figure 52: Workflow Canvas - Editor

Refer to the **Basic Operations** section to learn about the various basic operations performed in the **Editor** view.

Progress Bars

The **Progress Bars** indicate the size of the workflow and the number of nodes in the workflow. They are used to view the current status of the workflow and notify the user when the workflow reaches the maximum allowed size or complexity. The maximum allowed size and the number of nodes are 4.5 MB and 100 nodes, respectively. The workflow cannot be saved if it exceeds the maximum allowed size. If the workflow exceeds the maximum number of allowed nodes, it can be saved, but there may be performance issues when editing it.

The Progress bars are available in the Workflow Canvas as shown below:



Figure 53: Progress Bars

The Progress Bars represent the status of the workflow with the following colors:

Table 5: Progress Bars - Status

Color	Status
Green	Values are within the allowed limit, and there are no issues.

Color	Status
Yellow	The size or the number of nodes exceeds 50% of the allowed limit.
Red	The size meets or exceeds the allowed amount, or the number of nodes exceeds the recommended limit.

6.1.2 Running

The **Running** view shows all workflow instances that are currently running or have recently stopped. It is used to detect indefinitely running workflow instances and force them to stop.

- 1. On the **Workflow Canvas** page, click on the **Running** tab.

The **Running** view is displayed with the empty data.

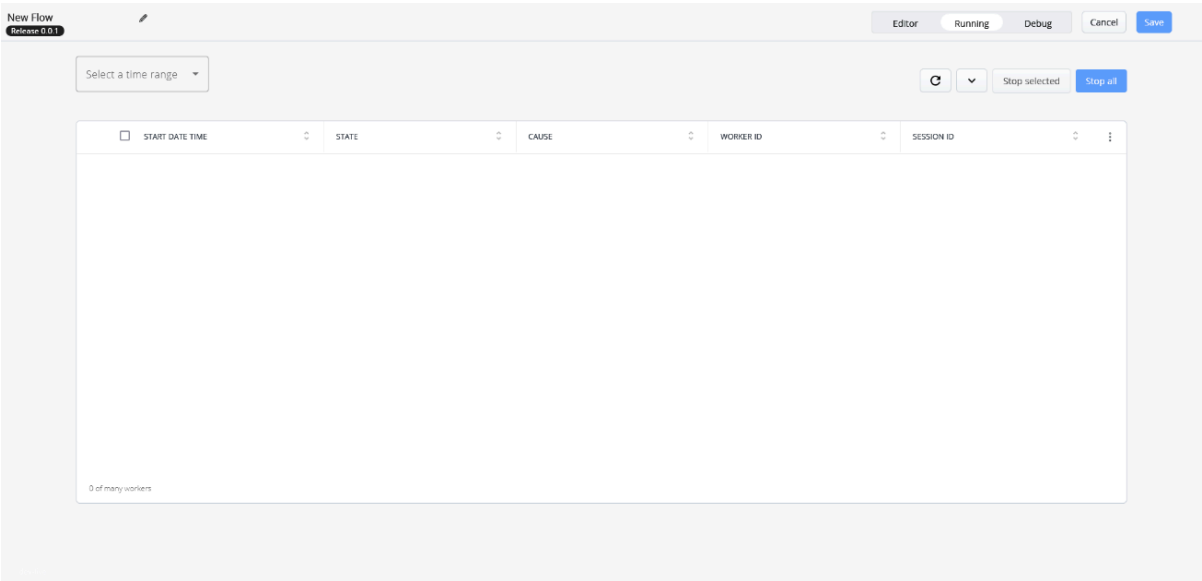


Figure 54: Workflow Canvas – Running

- 2. **Select a time range** from the drop-down list to filter the workflow instances for the selected time range.

The list of running or recently stopped workflows is populated with the workers.

For more information on columns, refer to the following table.

Table 6: Running View – Column Description

Columns	Description
Start Date Time	Displays the date and time that the workflow instance began running.


Columns	Description
State	<p>Displays the current state of the workflow instance.</p> <p>The available options are:</p> <ul style="list-style-type: none"> • Running: The workflow instance is actively running • Bookmarked: The workflow instance is waiting at a Bookmark activity • Cancelled: The workflow instance was terminated by an external request, such as stopping it from the Running view • Stopped: The workflow instance stopped normally
Cause	<p>Displays the reason for the current state, if any. This is generally populated with the Reason value in a Terminate action. It allows you to see why the workflow ended if a Terminate action is used to end the workflow.</p>
Worker ID	<p>Displays the unique identifier assigned to the instance of the worker.</p>
Session ID	<p>Displays the unique identifier assigned to the session.</p>

3.



Click on the up or down caret () icon in the column header to sort in ascending or descending order on any of the columns in the table.

4.

Click on the three dots () icon to filter the columns displayed in the table.

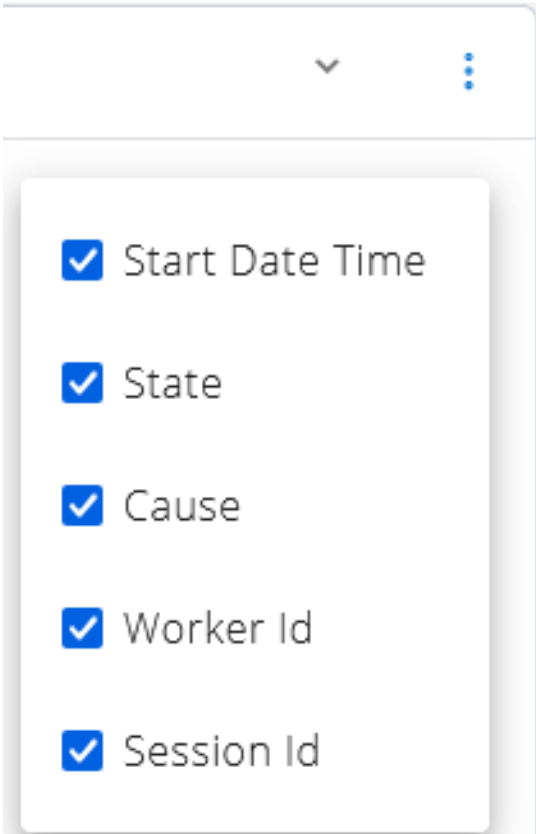


Figure 55: Running – Filter Parameters

5. Select the checkbox next to the column name to display it in the table.

6.




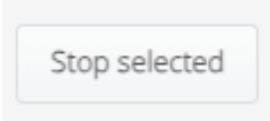
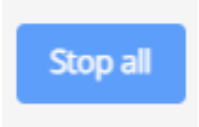
Click on the right arrow () icon to view the input and output variables for the workflow instance.



Figure 56: Running - Inputs/Outputs Variables

The following buttons are displayed at the top right of the page:

Table 7: Running View – Basic Actions

Buttons	Description
	Click this button to refresh the page for the currently selected time range. Note that the table will not automatically update. To see changes in workflow instances, you must either use the Refresh button or select another time range.
	Click this button to show all the Input and Output variables for all workflow instances in the table.
	Click this button to stop the selected workflow instance(s). This button is enabled only when one or more rows are selected and the workflow instance is in the Running state.
	Click this button to stop all the workflow instance(s). This button is enabled only when one or more workflow instances are in the Running or Bookmarked state.

i Note:

If a workflow instance is stopped in a **Bookmarked** state, it will remain in a **Bookmarked** state until the bookmark is resumed or times out. After that, it will be stopped, and the state will display as **Cancelled**.

6.1.3 Debug

After creating a workflow, you can enable debug to ensure the flow works as expected.

1. On the **Workflow Canvas** page, click the **Debug** tab to access the workflow debugger.

The **Debug Mode** pane is displayed on the right side.

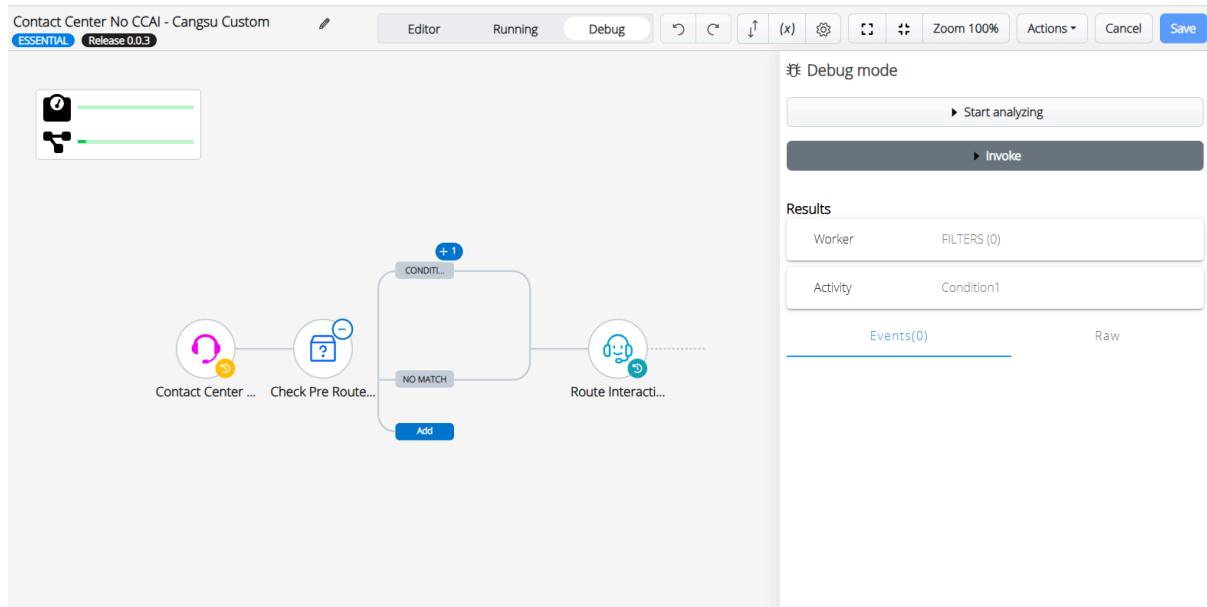


Figure 57: Debug

2. Select **Start analyzing** to begin analyzing and debugging the selected workflow.

The Debug information for each activity is displayed when your workflow is triggered.

For example,

- If your workflow begins with an **Endpoint Ringing** trigger, the workflow executes once the call is received from the selected endpoint.
- If your workflow begins with a **Webhook** trigger, click **Invoke** to execute the workflow. This will display a form where you can enter any input variables passed to your workflow when it starts.
- If there are no input variables, click **Run** on the form to trigger the workflow.

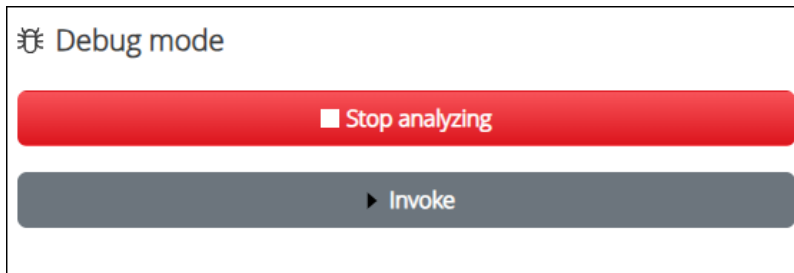


Figure 58: Debug mode

Once each activity is executed, a circled number appears with the total execution count. If the activity contains an error, the circle will be displayed in red.

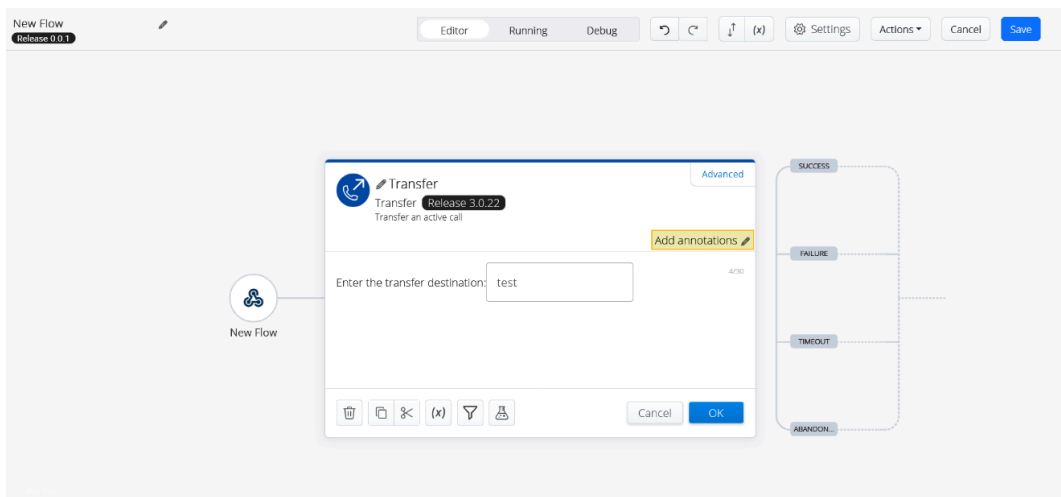


Figure 59: Circle Numbered Activities Sample

- After the workflow is complete, click **Stop analyzing** to stop the debugger and analyze the data from the workflow execution.

The **Debug mode** pane is displayed with the **Results** section.

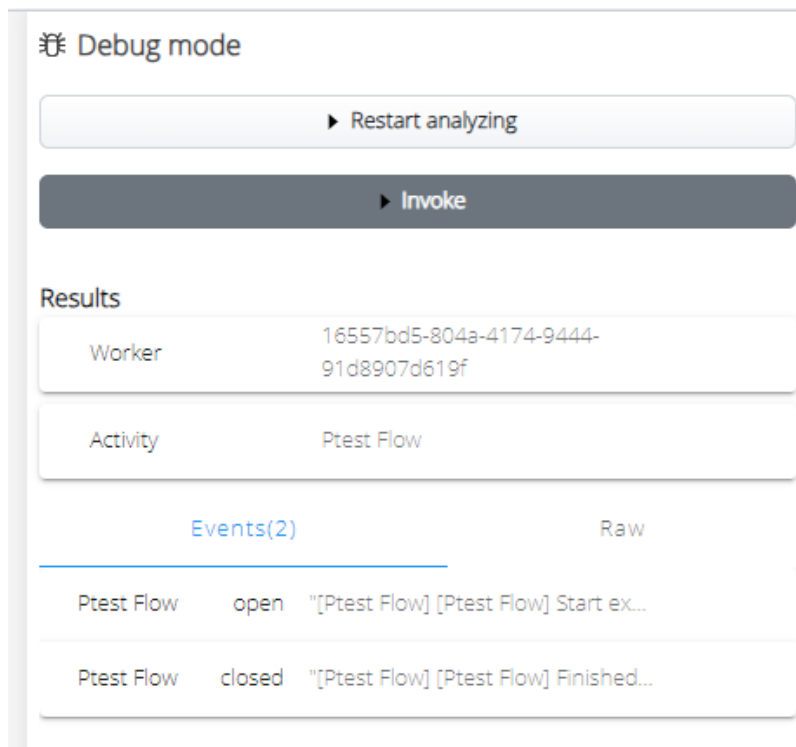


Figure 60: Debug mode - Results

For more information on the fields, refer to the following table below.

Table 8: Debug Results – Field Description

Field	Description
Worker	Displays the worker ID that executed this instance of the workflow. This is helpful when contacting Mitel support for further assistance or to look up additional information about the workflow's execution.
Results	
Events	Displays the information for each activity executed in the workflow. It is used for analysis. Each activity that is executed will have an Open and a Closed event.
Raw	Displays the HTTP messages received by the activity. It helps to see error messages or other detailed information returned in response.

- Click on the circled number on the activity to view more details.

The **Debug mode** pane is displayed with the activity details.

If the event type is **Open**, the **Events** section is displayed with all the input and throughput arguments of the activity.

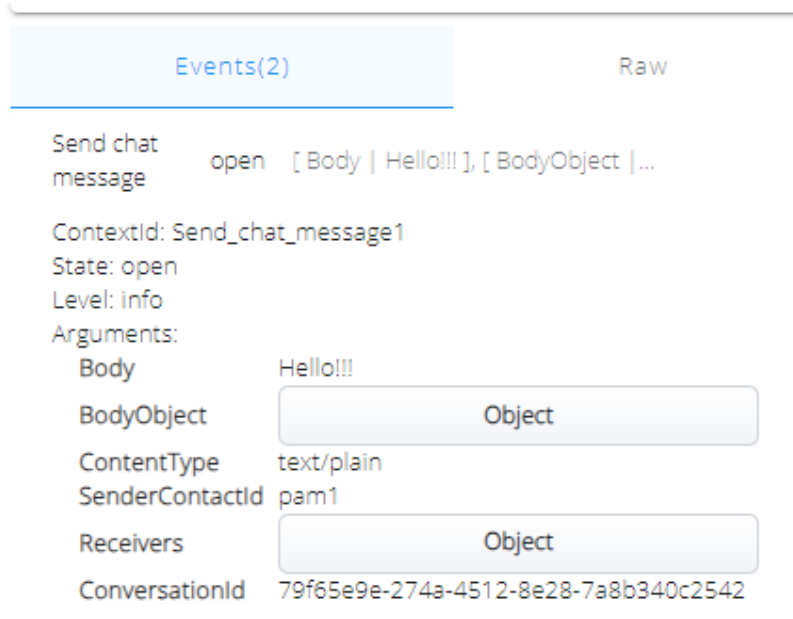


Figure 61: Debug mode – Open Events

If the event type is **Closed**, the **Events** section displays all the output and throughput arguments of the activity.

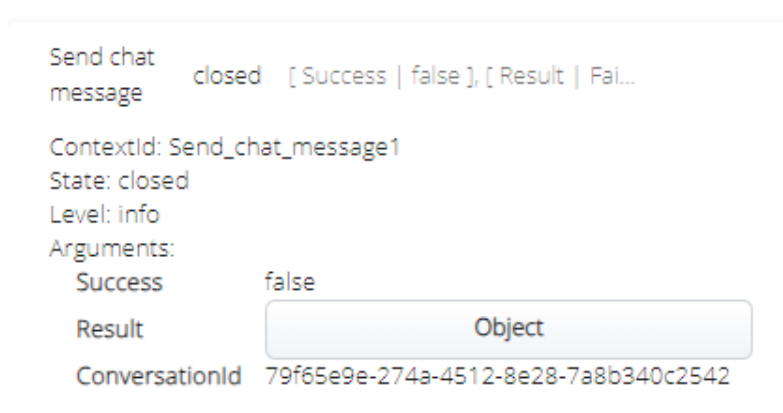


Figure 62: Debug mode – Closed Events

This information can be used to validate that each activity was executed as expected, and if not, it can be analyzed further.

- Click **Object** to view the details of the argument.

Debugging Multiple Activities:

You can also debug the workflow and its activities simultaneously. For example, if you debug the workflow and find that one of the activities contains an error, you can debug that activity at the same time as the workflow to obtain further information.

To debug an activity in a workflow, perform the following steps:

1. Click on the activity in the main workflow.
2. Click the **Advanced** tab.

The **Activity Identifier** is displayed below.

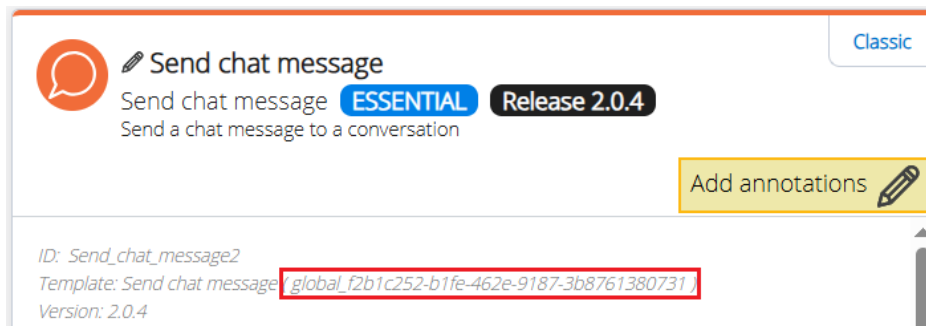


Figure 63: Activity Identifier

3. Click on the activity identifier to open that activity in a new browser tab.
4. In the **Workflow canvas**, click **Debug**.
5. Click **Start analyzing**.
6. Navigate to the main workflow and click **Start analyzing**.

When your workflow begins executing, the debug information is gathered for the main workflow and the selected activity.

7. Click **Stop analyzing** on both the main workflow and the selected activity tabs to stop debugging.

Note:

To debug one of the activities, the main workflow must be in active debug mode; it is impossible to start debugging the activity without debugging the main workflow.

This process can be repeated with multiple activities in the workflow simultaneously, so if any activity needs further detail, it can be debugged simultaneously with the main workflow.


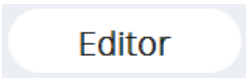
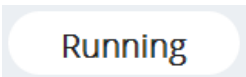
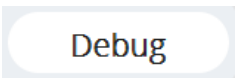




6.2 Basic Operations

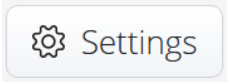


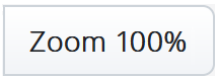
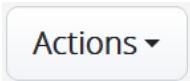
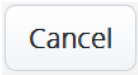
This topic describes the information on the basic operations performed in the Workflow Canvas.

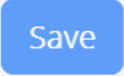
6.2.1 Building Workflows

Mitel Workflow Studio enables the CloudLink administrator to add new workflows or edit/delete existing workflows using the Workflow Canvas. As an administrator, you can perform the following actions within the Workflow Canvas:

Table 9: Workflow Canvas – Basic Operations

Operations	Description
	Click this icon to edit the name of workflow.
	Click this button to edit the existing activities on the workflow page.
	Click this button to view the execution history of the workflow.
	Click this button to troubleshoot issues.
	Click this icon to undo the changes made on the workflow page.
	Click this icon to redo the changes made on the workflow page.
	Click this icon to import/export the workflow.
	Click this icon to view variables-related options and select one of the following options: <ul style="list-style-type: none"> • Add variable • Delete variable • Search variables

Operations	Description
	Click this button to update the name, tags, and visibility of the workflow.
	Click this icon to expand all the nodes.
	Click this icon to collapse all the nodes.
	<p>Select the option from the list to change the display of the workflow.</p> <ul style="list-style-type: none"> • Zoom in • Zoom out • Zoom to fit • Zoom to 50% • Zoom to 100% • Zoom to 200% <p>Using this option, you can zoom in to view a particular section of a larger workflow.</p>
	<p>Click this button to select one of the following options:</p> <ul style="list-style-type: none"> • Set run flow as – Select this option to run the workflow as Owner, Caller, or Identity. • Invoke flow – Select this option to invoke the arguments to the selected workflow and run the results. • Export as SVG – Select this option to export the workflow as a scalable vector graphic file (.svg) for better presentation. • Clone this flow – Select this option to clone the workflow. • Delete this flow – Select this option to delete the workflow.
	Click this button to discard the changes.

Operations	Description
	Click this button to save the workflow.


6.2.2 Viewing Workflow

To navigate the Workflow Canvas, press and hold the mouse's left button anywhere on the workflow view and drag the mouse to display different areas of the workflow.

You can configure activities directly from the workflow page by clicking each one. Clicking an activity brings it into focus, enabling you to identify and update required changes in complex workflows quickly.

6.2.3 Import a Workflow


You can import a workflow by dragging and dropping, selecting a JSON file from your local folder, or copying and pasting the flow definition. By default, the **Create flow** option is used to create a new workflow with the imported workflow definition. If the definition is imported into the existing workflow, the imported workflow definition will override the current workflow.

1. On the **Workflow Canvas** page, click  icon.
The **Export/Import** popup screen is displayed.
2. In the **Import** tab, perform one of the following options to import the definition of the flow:
 - a. Drag and drop the JSON file containing the flow definition in the **Import from file** field.
 - b. Click **Select your json file** to select the JSON file from your local folder.
 - c. Copy and paste the flow definition under the **Paste here** section.
3. Click **Import**.

The selected workflow is imported to the **Workflow Canvas**.

6.2.4 Export a Workflow

You can export an existing workflow in JSON format.

1. On the **Workflow Canvas** page, click  icon.
The **Export/Import** popup screen is displayed.
2. In the **Export** tab, click **Export As** to export the flow in JSON format.
The selected workflow is downloaded to the local 'Downloads' folder.

6.2.5 Resizing Workflow

You can zoom in or out of the workflow using the mouse wheel. Scroll the mouse wheel up to zoom in, and scroll the mouse wheel down to zoom out. You can also use the Zoom option at the top of **Workflow Canvas** to change the display of the workflow.

6.2.6 Expanding/Collapsing Workflow

You can expand or collapse the nodes in the Workflow canvas. By default, the flow/action will be opened in collapsed mode.

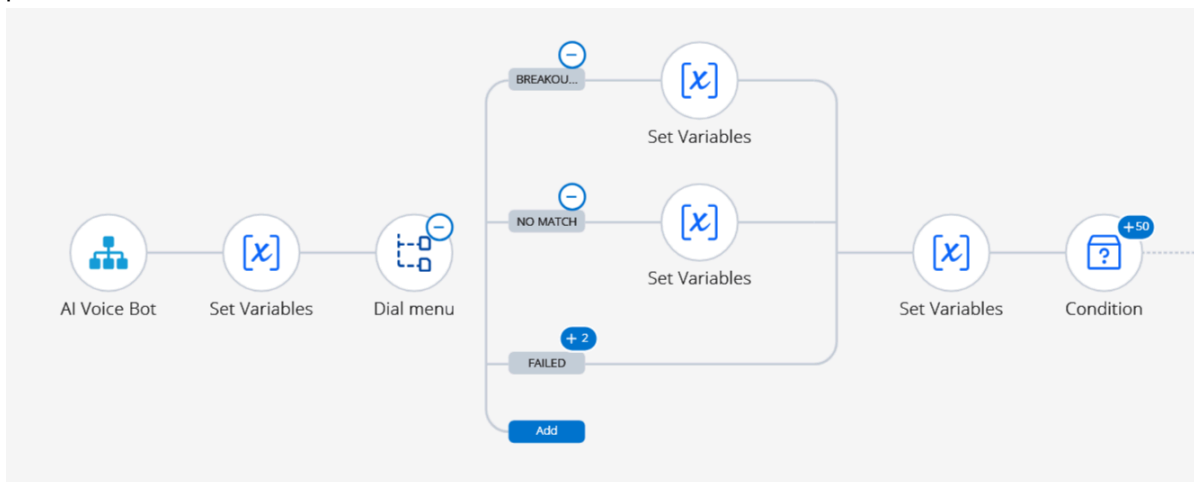



Figure 64: Workflow Canvas

The following actions can be performed in Workflow Canvas:

Table 10: Expand/Collapse Actions

Symbol/Icon	Action	Description
	Expand All	Click this icon at the top of the page to expand all the child nodes in the flow/action.
	Collapse All	Click this icon at the top of the page to collapse all the nodes in the flow/action and indicate the number of child nodes in the superscript number.
	Expand Node	Click the superscript number to expand all the child nodes for the selected node.

Symbol/Icon	Action	Description
	Collapse Node	Click the minus symbol to collapse all the child nodes and indicate the number of child nodes in the superscript number.

6.3 Activity Wheel

As an administrator, you can access the various activities under **Activity Wheel** from the **Workflow Canvas**.

The **Workflow Canvas** page displays the list of configured actions under **Activity Wheel**.

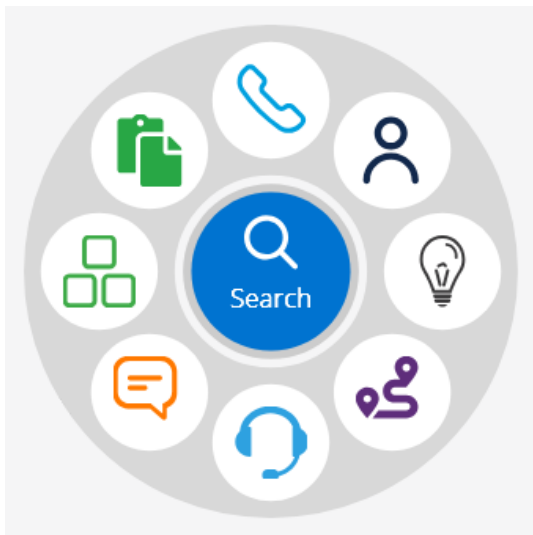


Figure 65: Activity Wheel







i Note:




For a detailed explanation of each activity, refer to the [List of Actions](#) on page 101 section.

6.3.1 Media Activities

The following table describes the details of the various Media activities supported in the **Workflow Canvas**:

Table 11: Media Activities




Activity Name	Icon	Description
Dial menu		This activity offers a caller a menu of options, prompting the caller to select service options by entering a configured digit or speaking a response. For example, "Press or say 1 for service and 2 for sales." The workflow will branch to different activities depending on the caller's selection.
Greeting		<p>This activity allows a caller to provide input by entering a digit or speaking a response, similar to the Dial menu activity.</p> <p>It allows the workflow to collect input from the caller and then process that input for cases that don't require a selection menu—for example, collecting speech from the caller and passing it to an AI Chatbot.</p>
Dial by name		This activity allows customers to find a specific employee in your company directory and then be automatically transferred to that person. Customers search the directory by entering two or more characters in the user's name through the customer's telephone keypad or by speaking the user's name. It is possible to exclude users such as the company CEO from the search results.
Transfer		<p>This activity transfers an active call to the provided destination.</p> <p>If the call is being transferred to a MiCC queue, data can be sent to MiCC along with it. See the Send MiCC Data activity for more information.</p>
Make call		This activity makes an outbound call to the configured destination from another device. Passing in an alternate From Name to the Make Call activity can override the calling party name.
Answer		This activity answers an inbound, ringing call. It is typically used with an Endpoint Ringing trigger.

Activity Name	Icon	Description
Hang up call		This activity ends an active call being handled by the workflow.
Record audio message		This activity lets the caller record a short audio message for the configured number of seconds. A beep tone will play, and the caller can record a message. The activity can be configured to hang up the call after the recording completes or allow the call to continue.
Get call tag		Voice calls can be tagged with identifying information, such as data about the caller, that is looked up and attached to the call. This activity retrieves data for a specific call, which can be used in the workflow.

6.3.2 User Management Activities

The following table describes the details of the various User Management activities supported in the **Workflow Canvas**:







Table 12: User Management Activities








Activity Name	Icon	Description
Get user		This activity retrieves a CloudLink user based on the user ID, email address, or mobile number.
Search users		This activity retrieves one or more CloudLink users or PBX contacts matching a provided search term, such as a name or mobile number. The search term can be characters corresponding to a dial keypad or the actual name or number value. An array of matching users and contacts are returned.
User presence		This activity retrieves or sets CloudLink presence for one or more selected users. Presence can be set to any predefined values, and an optional reason can be provided. The retrieved or set presence values are returned in an array.






6.3.3 Developer Activities

The following table describes the details of the various Developer activities supported in the **Workflow Canvas**:

Table 13: Developer Activities

Activity Name	Icon	Description
Condition		This activity configures conditions that branch the workflow, adds one or more branches, and groups multiple branches. This allows the workflow to take different actions depending on the customer's actions or the data received.
Set variables		This activity sets the value of variables in the workflow. For more information, refer to the Variables topic.
HTTP request		This activity allows you to configure the HTTP activity to access RESTful Web APIs in your workflow. You can send requests using HTTP to web services to retrieve or update data. The HTTP activity encapsulates the header and body of the API request, making it easy to use within your workflow.
Log		This activity adds log messages, which provide additional information for debugging a flow.
Terminate		This activity allows you to stop the workflow execution completely.
For each		This activity allows the workflow to iterate through an array and execute activities on each item.




Activity Name	Icon	Description
Go to		This activity allows you to redirect the flow of a workflow. The workflow will continue executing the activities as indicated in this activity.
Break		This activity breaks the workflow from the current logical flow. This is commonly used with a For Each activity to stop iterating through the array and continue the workflow before all items are evaluated.
Delay		This activity allows you to delay or pause the workflow for a specified number of milliseconds (1000 milliseconds = 1 second).
Continue		This activity breaks from the current logical flow and continues at the next iteration of a loop. It is commonly used with a For Each activity to stop the current iteration and continue with the next entry in the array.
Bookmark		This activity pauses the workflow for a specified time while waiting for an action. If the action does not occur before the configured timeout value, the workflow resumes.
Open session		This activity opens a new session for the workflow based on an identifier, such as a call ID or session ID. This is typically used in conjunction with a Bookmark activity. When the Open Session activity is executed, it will check if an existing session exists for the workflow running with the same identifier. If so, that session is resumed, and the existing Bookmark exits. Otherwise, a new session is created.
JavaScript		This activity supports native JavaScript functions implemented directly in the JavaScript engine and does not require additional libraries. It can perform basic functions such as formatting or parsing data. The activity has access to all the workflow variables and can return a result to the workflow, along with additional data, when it is complete.

Activity Name	Icon	Description
Cloudlink subscribe		If the workflow requires CloudLink notifications, the Subscribe activity can specify the topic, subject, and method, along with a filter for specific notification events. For example, to monitor the state of a call at an endpoint, the workflow can use this activity to receive notifications from platform-api-media for that specific endpoint and call. Refer to the CloudLink API documentation for further details.
Cloudlink unsubscribe		This activity cancels a subscription set up for CloudLink notifications through a Subscribe activity. The Subscribe activity returns the Subscription ID(s), which is passed to the Unsubscribe activity to stop receiving the requested CloudLink notifications.
Parallel		This activity executes two or more branches in a workflow simultaneously. Each branch executes simultaneously, and the workflow waits for all branches to complete or one of the branches to complete, depending on the configuration of the Parallel activity.
Try-Catch		This activity creates two branches: the "Try" branch will execute, and the workflow will continue with the "Catch" branch if an error is encountered. This allows you to add error handling to your script.
While		<p>This activity causes a workflow to repeat activities until a condition is met. For example, while the variable count < 3, continue to repeat an activity and increment count. It can be executed in two different modes. "While" mode evaluates the condition first, and the loop is not executed if it is already met. "Do While" mode executes the loop and then evaluates the condition, so it is guaranteed to execute it at least once.</p> <p>This activity is used with a Continue or Break activity to return to the While activity and evaluate the condition or break out of the While loop. This activity should be used carefully so that an infinite loop is not created.</p>

6.3.4 Routing Activities

The following table describes the details of the various Routing activities supported in the **Workflow Canvas**:

Table 14: Routing Activities



Activity Name	Icon	Description
Schedule		<p>This activity allows you to set the working hours and holiday schedules for your organization.</p> <p>If your account has Google integration, the Google Places API can automatically retrieve the working hours of the selected organization.</p>
Emergency mode		<p>This activity turns on or off emergency mode for your account, allowing you to route interactions differently based on the configured setting.</p>
Phone number lookup		<p>This activity allows you to look up a phone number to determine the approximate location attributes of the caller.</p>






6.3.5 MCX Activities

Prerequisites: CloudLink Daemon is required to run the MCX Activities.

The following table describes the details of the various MCX activities supported in the **Workflow Canvas**:

Table 15: MCX Activities







Activity Name	Icon	Description
MCX get employee state		<p>This activity retrieves the current aggregate state for a MCX employee and an optional reason. The state can be any of the following values: Unknown, Available, Busy, Do Not Disturb, Away, or Offline.</p>
MCX change employee state		<p>This activity sets the aggregate state for a MCX employee with an optional reason. The state can be any of the following values: Unknown, Available, Busy, Do Not Disturb, Away, or Offline.</p>



Activity Name	Icon	Description
MCX get queue state		This activity retrieves the current real-time statistics for a MCX queue, including the number of available agents, the number of idle agents, the number of waiting conversations, and whether the queue is in a Do Not Disturb state.
MCX port state		This activity sets or retrieves the current emergency setting of a MCX hunt group, extension, or media server. If the device is in emergency mode, alternate routing choices can be used in MCX, such as when there is an extreme weather situation.
MCX route open media		This activity sends an Open Media request to the MCX system to be routed to an MCX agent. The request can be any type of data object, such as a social media message.
MCX send data		This activity sends selected variables to MCX when a call is transferred. It allows the user to select which of the workflow variables will be sent to MCX, and then, when a call is transferred, the selected variables are sent along with the call.
MCX callback		This activity adds a callback to the MCX system, which can be routed to an MCX agent. The customer's name and number can be specified, as well as the callback destination number. It is required to specify the MCX media server that will handle the callback, as well as the subroutine in MCX that will handle the callback routing.

6.3.6 Messaging Activities

The following table describes the details of the various Messaging activities supported in the **Workflow Canvas**:

Table 16: Messaging Activities


Activity Name	Icon	Description
Create chat stream		This activity creates a new chat stream for a chat conversation.
Send chat message		This activity sends a chat message to all users in an existing chat conversation or creates a new conversation to send a direct message to selected users. The content type of the message is commonly text or HTML, but other advanced options are available, including Action Card, Location, Missed Call, Voicemail, or System Message.
Ask		This activity sends a question to an existing chat conversation and returns the answer received. A timeout value can be configured to complete the activity if an answer is not received within the timeout period (default is 60 seconds).
Add participant		This activity adds one or more participants to an existing chat conversation. The users can be added as guests or full participants. If the participant is not added as a guest, it is validated as a CloudLink user or client.
Tag chat		This activity adds tags to an existing conversation. Tags can be used to categorize chat conversations.
Delete chat		This activity deletes an existing chat conversation. Messages will no longer be accepted for chat conversations.




Activity Name	Icon	Description
Create chat transcript		<p>The contents of an existing chat conversation can be captured in HTML format. This activity allows the number of days to be selected for the transcript.</p> <p>Note: The workflow creator must be a member of the conversation to use this activity.</p>
Create or update attachment		<p>This activity allows the workflow to add a new attachment to a chat conversation or update an existing attachment. The name and type of attachment can be specified. A URL can be configured to retrieve the attachment. It can be specified in the activity in a Base64 encoded buffer.</p> <p>Note: The workflow creator must be a member of the conversation to use this activity.</p>





6.3.7 Integration Activities


The following table describes the details of the various Integration activities supported in the **Workflow Canvas**:




Table 17: Integration Activities






Activity Name	Icon	Description
Office 365 Integration For Outlook Integration procedure, refer to the Outlook Integration (Essential) on page 288 section.		
Office 365 Outlook – Send email		<p>This activity sends an email through Office 365 to selected email addresses. A connection must first be configured to Microsoft Office 365 – Outlook from Workflow to use this activity. The email content, recipients (including CC and BCC recipients), and email attachments can be specified.</p>





Activity Name	Icon	Description
Twilio Integration For Twilio Integration procedure, refer to the Twilio Integration (Premier) on page 300 section.		
Twilio send message		This activity sends an SMS message from a Twilio account to a specified number. To use this activity, a connection must be configured to Twilio.
CM.com Integration For CM.com Integration procedure, refer to the CM.com Integration (Premier) on page 232 section.		
CM.com send message		This activity sends an SMS message from a CM.com account to a specified number. A connection must be configured to CM.com to use this activity.
OpenAI Assistants Integration For OpenAI Assistants Integration procedure, refer to the OpenAI Assistants Integration (Premier) on page 282 section.		
ChatGPT		<p>For an existing OpenAI connection, this activity can send a message to the ChatGPT chatbot and receive a response. The message to be sent and the model to interpret the message are input values to the activity. For details on the supported model options, refer to the OpenAI documentation. The activity returns the response as a string; the full response is in JSON format.</p> <p>To use this activity, a connection must be configured to OpenAI.</p>






Activity Name	Icon	Description
OpenAI Assistants		<p>This activity enables the workflow to use the OpenAI Assistants API to build an artificial intelligence (AI) assistant into your workflow. It allows instructions to be sent to the AI, as well as requests which the AI assistant responds to. For more details, refer to the OpenAI Assistants API documentation.</p> <p>To use this activity, a connection must be configured to OpenAI.</p>
OpenAI Text-to-speech		<p>This activity plays a message using one of the OpenAI text-to-speech voices. If the speech is collected from the caller, it will wait for the user to respond and return the response from the user in the activity response output variable.</p>
OpenAI Speech-to-text		<p>This activity transcribes an audio file and has the ability to translate the audio file into OpenAI supported languages. For more details, refer the OpenAI documentation.</p>
Microsoft Graph Integration <p>For Microsoft Graph Integration procedure, refer to the Microsoft Graph Integration (Premier) on page 268 section.</p>		
Teams find user		<p>This activity can be used for an existing Microsoft Graph API connection to look up a user defined on the Microsoft Azure system. A filter is passed to the activity as an input value to provide search details for the user, such as mail ID or User Principal Name. If a user is found matching the provided filter, the activity returns a JSON-formatted response with the user's details.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>


Activity Name	Icon	Description
Teams presence		<p>For an existing Microsoft Graph API connection, this activity can be used to set, retrieve or clear the presence of a user defined on the Microsoft Azure system. The user's ID is an input value to the activity, along with an action value indicating Set, Get or Clear presence. If Set presence is requested, the formatted body of the request must also be provided to the activity to specify the presence value, as shown below:</p> <pre>{ "sessionId": "22553876-f5ab-4529-bffb-cfe50aa89f87", "availability": "Available", "activity": "Available", "expirationDuration": "PT1H" }</pre> <p>Refer to the Microsoft Graph REST API documentation for details on the options for setting presence.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>

Activity Name	Icon	Description
Send message to MS Teams channel		<p>For an existing Microsoft Graph API connection, this activity can be used to send a message to an existing channel (or conversation) for a team. The Team ID, Channel Name, and Message content are input values to the activity. The Message must be sent in JSON format. An example is shown below:</p> <pre>{ "body": { "content": "Hello World" } }</pre> <p>Refer to the Microsoft Graph REST API documentation for additional details.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>
OneDrive - File upload		<p>This activity uploads a file to a specified location on One Drive.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>
OneDrive - Get file contents		<p>This activity allows the workflow to retrieve the contents of a .pdf, .txt, or .docx file that exists on a One Drive account.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>

Activity Name	Icon	Description
Excel get table rows		<p>This activity allows the workflow to retrieve data from a table in an Excel workbook.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>
Excel add, update or delete table rows		<p>This activity allows the workflow to add, update or delete data in a table defined in an Excel workbook.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p> <p>A connection must be configured to Microsoft Graph API to use this activity.</p>
Get calendar availability		<p>This activity looks up a specific date and time period on the user's calendar in the Microsoft Azure system to determine if the user is available. If the user is not available, alternate times can be returned.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p>
Set calendar event		<p>This activity sets a new event in the user's calendar on the Microsoft Azure system for a specific date and time period.</p> <p>For Azure permissions, refer to the Microsoft Azure Integration section.</p>
Google Integration <p>For Google Integration procedure, refer to the OpenAI Assistants Integration (Premier) on page 282 section.</p>		
Google Drive - File upload		<p>This activity uploads a file from a shared URL to a location specified on a Google Drive.</p>

Activity Name	Icon	Description
Google Drive - Get file contents		This activity returns the contents of any selected files that exist in a Google Drive account. The supported files are .pdf, .txt or .docx.
Salesforce Integration For Salesforce Integration procedure, refer to the Salesforce Integration (Premier) on page 296 section.		
Salesforce create, update or delete record		<p>This activity allows the workflow to create, update, or delete a record from a Salesforce database. The record type, record ID, and request information are specified in the activity.</p> <p>To use this activity, a connection must be configured to Salesforce.</p>
Salesforce query SOQL		<p>This activity enables access to the Salesforce Object Query Language (SOQL) from the workflow. This allows Salesforce data to be searched for specific information, similar to the SELECT statement in SQL. For more details, refer to the Salesforce Object Query Language documentation.</p> <p>To use this activity, a connection must be configured to Salesforce.</p>
Salesforce get record field values		<p>This activity allows the workflow to retrieve field values for specific Salesforce data. The record ID of the data, along with the fields to be retrieved, is specified.</p> <p>To use this activity, a connection must be configured to Salesforce.</p>
Google Gemini Integration For Gemini Integration procedure, refer to the Google Gemini Integration (Premier) section		

Activity Name	Icon	Description
Google Gemini		<p>This activity enables the workflow to use the Google Gemini artificial intelligence chatbot. It provides an easy-to-use interface where instructions can be sent to the chatbot, along with a message. The output of the activity is the response from the chatbot.</p> <p>To use this activity, a connection must be configured to Google Gemini.</p>
Anthropic Integration For Anthropic Integration procedure, refer to the Anthropic Integration (Premier) section.		
Anthropic Claude		<p>This activity enables the workflow to use the next-generation artificial intelligence (AI) assistant Claude from Anthropic. It provides an easy-to-use interface where instructions can be sent to the AI, along with a message. The output of the activity is the response from the AI assistant.</p> <p>To use this activity, a connection must be configured to Anthropic.</p>
Jira Integration For JIRA Cloud Integration procedure, refer to the Jira Cloud Integration (Premier) on page 264 section.		
Jira add issue comment		This activity adds a comment to an existing Jira issue
Jira create issue		This activity creates a new Jira issue in the indicated project with the provided summary and description. The issue can be assigned to a Jira user, and a custom field can also be set.
Jira get issue		This activity retrieves an issue from Jira

Activity Name	Icon	Description
Jira update issue		This activity updates the summary and description for an existing Jira issue.


**Note:**

The activities appear under **Integrations** based on the applications currently integrated into Workflow Studio. Refer to the **Integration** topic for more details on integrating an application into Workflow Studio.

6.3.8 Paste Activity

As an administrator, you can copy and paste any activity to any preferred location in the workflow.

Table 18: Paste Activity

Icon	Description
	This activity allows you to copy and paste any activity to any preferred location in the workflow.

6.4 Expressions

The workflow expression language is used anywhere there is an assignment to a variable or for constructing comparisons. For example, the following expression ensures that these conditions are met:

- The value of the string variable named “membership” is set to “gold”.
- The value of the numeric variable named “waitTime” multiplied by the numeric variable named “factor” is greater than 4000 OR The value of the string variable named “status” is set to “pending” or “error”.

```
membership == "gold" and (waitTime * factor > 4000 or status in ("pending", "error"))
```

6.4.1 Values

There are five supported value types for variables.

Examples of values for different variable types are described in the following table.

Table 19: Values

Variable Type	Value
Boolean	True, False
Number	45, -90.3, NaN, Infinity
String	abcd, 23cvb54R 3rt4+53yt
Object	someObject
Array	(a, b, c) someArray

6.4.2 Variables

```
variableName
```

```
variableA + variableB
```

```
anotherVariable != False
```

All variables present in the current expression scope are available for use. This includes:

- User-defined variables within the workflow
- Arguments of all activities in the workflow

Variables are accessible via their plain text name and will be displayed as autocomplete options. You can enter the first few characters of the variable, and the system will complete the name with the accessible variables for the workflow.

Variables will always be of one of the five Value types:

- Boolean, Number, String, Object, Array

6.4.3 Arithmetic

```
( x + 5 ) ^ ( y % 3 ) - sqrt( x )
```

All common arithmetic operators are supported:

- Addition, Subtraction, Multiplication, Division
 - $x + y$
 - $x - y$
 - $x * y$
 - x / y
- Exponentiation
 - $x ^ y$
- Modulus - Returns the remainder from integer division.
 - $x \% y$

Several useful arithmetic functions are also available:

- Absolute Value
 - `abs(x: number): number`
- Natural Logarithm
 - `log(x: number): number`
- Square Root
 - //Equivalent to $x ^ (1 / 2)$
 - `sqrt(x: number): number`
- Ceiling - Rounds upwards to the nearest integer
 - `ceil(x: number): number`
- Floor - Rounds downwards to the nearest integer
 - `floor(x: number): number`
- Minimum - Returns the smallest number in the given list.
 - `min(a, b, c...): number`
- Maximum - Returns the largest number in the given list.
 - `max(a, b, c...): number`
- Rounding - Round to the nearest integer
 - `round(x: number): number`
- Random - Random floating-point number from 0.0 to 1.0
 - `random(): number`

6.4.4 Boolean

```
( x or y ) and not z
```

The following Boolean operators are supported:

- AND - TRUE if both x AND y are TRUE - FALSE otherwise.

```
x and y
```

- OR - TRUE if x OR y is TRUE OR both are TRUE - FALSE otherwise.

```
x or y
```

- NOT - TRUE if x is FALSE - FALSE otherwise.

```
not x
```

- Ternary - If x is TRUE, return y; otherwise, return z.

```
x ? y : z
```

6.4.5 Equalities

```
volume > 56
```

```
color != "green"
```

The following equality operators are supported:

- Equal, Not Equal
- Less than, Less than or equal
- Greater than, Greater than or equal

6.4.6 Arrays

Arrays are lists of values.

The following table describes the list of functions available for Arrays:

Table 20: Arrays

Expression	Description
<code>ArrayCreate(...elements): []</code>	Returns an array with the provided elements.
<code>ArrayGetItem(array: Array, index: number)</code>	Returns the item at the indicated index value in the array. Note that the first element will have an index value of 0.

Expression	Description
<code>ArrayConcat(array: Array, ...values)</code>	Return a new array that adds the provided values to the end of the existing array.
<code>ArrayCount(array: Array)</code>	Returns the number of elements in the array.
<code>ArrayJoin(array: Array, separator: string)</code>	Returns a string with all the elements in the array. If the separator is provided, it is returned between each array element.
<code>ArrayFilter(array: Array, expression: string)</code>	Applies the expression provided against all elements in the array and returns a new array with only elements returning true when evaluated against the expression.
<code>ArrayFind(array: Array, expression: string)</code>	Returns the first element in the array, which returns true when evaluated against the expression.
<code>ArrayIncludes(array: Array, expression: string): Boolean</code>	Returns true if the array includes an element that returns true when evaluated against the expression. Otherwise, returns false.
<code>ArraySlice(array: Array, start: number, end: number): Array</code>	Returns a new array starting at the index provided in the start and ending at the index before the value provided in the end or until the end of the array is reached. If an end is not provided, all elements from the start index until the end of the array are returned.
<code>ArrayInsert(array: Array, items: Array, start: number): Array</code>	Insert the items into the array at the position indicated by the start. If no start value is provided, the items are appended to the end of the array.

Creation

There are two ways to create an array. The array can be directly assigned with members, as in the following examples:

```
array1 = (a, b, c)
```

```
array2 = (True, "abc", 5)
```

The function `ArrayCreate` can also be used to create an array, as in the following examples:

```
array3 = ArrayCreate(a, b, c)
```

```
array4 = ArrayCreate(True, "abc", 5)
```

**Note:**

The arrays can contain items of different value types.

Membership

```
x in (a, b, c)
```

```
x not in (a, b, c)
```

It is possible to check whether an element is included in an array using the `in` operator. For example, `x in (a, b, c)` will return `True` if the value of variable `x` is `a`, `b`, or `c`. Otherwise, it will return `False`.

Using `not in` provides the reverse functionality, where `True` is returned if the value of the variable is not in the array; otherwise, `False` is returned.

Element Access

There are two ways to access elements of an array. The element can be directly accessed by the index value, such as `myArray[2]`. If the array is defined as `myArray = (a, b, c)`, `myArray[2]` has the value of `c`. Note that array indices begin with 0.

The function `ArrayGetItem` can also be used to access an element of an array. The following example returns the third element in `myArray`: `ArrayGetItem(myArray, 2)`

Appending items

Elements can be appended to an array using the function `ArrayConcat`. Multiple items can be added at the same time. The following example adds `d`, `e`, and `f` to `myArray`: `ArrayConcat(myArray, d, e, f)`

Length

The function `ArrayCount` returns the length of the array.

```
ArrayCount(array: Array(any)): number
```

6.4.7 Objects

Generic Objects are supported and are bags of Key-Value pairs, where a key points to a specific Value.

The following table describes the list of functions available for Objects:

Table 21: Objects

Object Name	Expression	Description
Creation	<ul style="list-style-type: none"> ObjectCreate(obj: object, property: string) ObjectCreate(someObject, "property1")) 	<p>Example:</p> <pre>ObjectCreate(("count", "color", "isReady"), (23, "red", true))</pre> <p>The <i>ObjectCreate</i> creates an object with three keys (count, color, and isReady) with the following values: 23, "red" and True. The representation in JSON notation is as follows:</p> <pre>{ "count": 23, "color": "red", "isReady": true }</pre> <p>Use the <i>ObjectCreate</i> function to create objects. One or more keys and a corresponding number of values can be passed.</p>
Member Access	<ul style="list-style-type: none"> x.y y of x 	The key attribute can be used to access members of the object. Using the example above, <i>myObject.count</i> = 23, and <i>myObject.color</i> = "red"
Check the existence of object property	<ul style="list-style-type: none"> ObjectHasProperty(obj: object, property: string) ObjectHasProperty(someObject, "property1")) 	The function <i>ObjectHasProperty</i> can be used to check if a key exists in an object. It will return True if the key exists in the object; otherwise, it returns False.
Set the object property	ObjectSetProperty(obj: object, propertyName: string, propertyValue: any): object	The function <i>ObjectSetProperty</i> can be used to set the indicated value to the propertyValue provided and return the updated object.

Object Name	Expression	Description
Convert an object to a string	<code>ObjectStringify(obj: object): string</code>	The function <i>ObjectStringify</i> can be used to convert an object in JSON format to a string.
Assign the object values	<code>ObjectAssign(targetObj: object, sourceObj: object): object</code>	The function <i>ObjectAssign</i> can be used to assign all values in sourceObj to targetObj and returns targetObj.
Search an object	<code>ObjectQuery(obj: object, expression: string)</code>	The function <i>ObjectQuery</i> can be used to search for an item matching the expression in the object and return it. If the expression starts with \$, a JSON query is made on the object. Otherwise, the expression is applied to elements of the array or object to find a match.
Convert a JSON string to an object	<code>JSONParse(objString: string): object</code>	The function <i>JSONParse</i> can be used to convert a JSON string into an object and returns the object.
Return a GUID	<code>UUID()</code>	The function <i>UUID</i> can be used to return a GUID value.

6.4.8 String Expressions

Several convenience functions are available for common String processing situations.

The following table describes the list of functions available for String expressions

Table 22: String Expressions

Type	Expressions	Description
Length	<code>StringLen(str: string): number</code>	Returns the length of the string.

Type	Expressions	Description
Conditional	<ul style="list-style-type: none"> StringStartsWith(str: string, subStr: string): Boolean Returns True if the string begins with the provided substring; otherwise, it returns False. StringContains(str: string, subStr: string): Boolean Returns True if the substring is contained anywhere in the string; otherwise, it returns False. StringEndsWith(str: string, subStr: string): Boolean Returns True if the string ends with the provided substring; otherwise, it returns False. 	Returns the conditional value of the function. True or False.
Building	StringBuilder(...strings): string	Concatenates multiple strings into a single string
Trim	StringTrim(str: string)	Removes all white space characters from the beginning and end of the string.
To Lower Case	StringToLowerCase(str: string)	Converts the entire string to lowercase.
Regular Expressions	<ul style="list-style-type: none"> x ~= y x ~= ^(abc) 	These are patterns used to match character combinations in strings. ~= indicates that the string is being compared to a regular expression and will return True if the string matches the expression or False if it does not. Refer to RegEx documentation for pattern descriptions.
Index Of	StringIndexOf(str: string, subStr): Number	Returns the index of subStr in str, or -1 if subStr is not contained in str.

Type	Expressions	Description
Replace	<code>StringReplace(str: string, replaceStr: string, withStr: string): string</code>	Searches for the value <code>replaceStr</code> in <code>str</code> and replaces it with the value <code>withStr</code> . For example, <code>StringReplace('abcdefg', 'abc', 'xyz')</code> returns the string 'xyzdefg'.
Replace Regular Expression	<code>StringRegexReplace(str: string, regexp: string, withStr: string, flags?: string): string</code>	Searches for the regular expression represented by <code>regexp</code> in <code>str</code> and replaces it with the value in <code>withStr</code> . The flags can be used to designate options when using the regular expression. Refer to RegEx documentation for more details on using regular expressions.
Return a new string	<code>StringSlice(str: string, start: number, end: number): string</code>	Returns a new string starting at the character provided in the <code>start</code> and ending at the index before the value provided in the <code>end</code> or until the end of the string is reached. For example, <code>StringSlice('abcdefg', 2, 5)</code> returns the string 'cde'. If an <code>end</code> is not provided, the substring continues until the end of <code>str</code> .
Split the existing string	<code>StringSplit(str: string, separator: string): [string]</code>	Splits the existing string into an array of strings based on the provided separator character. For example, <code>StringSplit('a+b+c', '+')</code> returns the array ['a', 'b', 'c'].
Match the regular expression	<code>StringMatch(str: string, regexp: string, flags: string): [string]</code>	Searches for the regular expression provided in <code>regexp</code> in <code>str</code> and returns an array of all matches. The flags can be used to designate options when using the regular expression. Refer to RegEx documentation for more details on regular expressions.
Convert to Base64 format	<code>StringToBase64String(any): string</code>	Converts any input value to a string in Base64 format.
Convert from Base64 string to UTF-8 encoding	<code>StringFromBase64String(any): string</code>	Converts a value from a Base64 string to a regular string in UTF-8 encoding.

Type	Expressions	Description
Encode URI component	<code>StringEncodeURIComponent(str: string): string</code>	Encodes a string as a valid Uniform Resource Identifier (URI) by escaping characters requiring this.
Decode URI component	<code>StringDecodeURIComponent(str: string): string</code>	Returns the unencoded version of an encoded Uniform Resource Identifier (URI) by unescaping characters.

6.4.9 Date and Time Expressions

Thu Jan 26 2021 11:00:00

While there is no dedicated Date or Time type, the following table describes the several convenience functions for working with Date and Time Strings:

Table 23: Date and Time

Expression	Description
<code>DateUtcNow(): string</code>	Returns the current date and time in Coordinated Universal Time (UTC).
<code>DateUtcAddDay(date: string, daysToAdd: number): string</code>	Returns the date/time by adding the specified number of days to the provided date/time.
<code>DateUtcAddHour(date: string, hoursToAdd: number): string</code>	Returns the date/time by adding the specified number of hours to the provided date/time.
<code>DateUtcAddMinutes(date: string, minutesToAdd: number): string</code>	Returns the date/time by adding the specified number of minutes to the provided date/time.
<code>DateUtcAddSeconds(date: string, secondsToAdd: number): string</code>	Returns the date/time by adding the specified number of seconds to the provided date/time.
<code>DateUtcSubDay(date: string, daysToSub: number): string</code>	Returns the date/time by subtracting the specified number of days from the provided date/time.

Expression	Description
<code>DateUtcSubHour(date: string, hoursToSub: number): string</code>	Returns the date/time by subtracting the specified number of hours from the provided date/time.
<code>DateUtcSubMinutes(date: string, minutesToSub: number): string</code>	Returns the date/time by subtracting the specified number of minutes from the provided date/time.
<code>DateUtcSubSeconds(date: string, secondsToSub: number): string</code>	Returns the date/time by subtracting the specified number of seconds from the provided date/time.
<code>DateDifference(date1: string or number, date2: string or number, duration: string): number</code>	Returns the difference between the two dates or numbers based on the duration value, where the duration can be any of the following: seconds, minutes, hours, days, weeks, months, or years.
<code>DateGetTime(): number</code>	Returns the current UTC in milliseconds.
<code>DateUtc(): string</code>	Returns the current UTC formatted as a string.
<code>DateIso(date: string, number or Date): string</code>	Returns the provided date value formatted as a string in ISO format.

6.5 Using Variables

Variables are used in workflows, actions, and triggers to perform various tasks, including:

- Storing information for later use
- Collecting user data
- Performing database queries
- Building simple or complex conditions

Variables can be used within different activities, such as:


- Transferring a call to a different destination based on customer class.
- Playing different prompts based on account balance.
- Holding collected digits for use later in a database query.

Some variables can also be passed from Workflow Studio to users as a desktop screen pop, providing them with customer information gathered by the workflow.

Variables may be configured to mask their contents, enabling the variable to be displayed either entirely masked or partially masked, with a user-specified number of unmasked characters (`***1234`). This limits the exposure of potentially sensitive information, such as credit card or personal identification numbers, in logs, call notes, databases, and screen pops. Secured variables cannot be used with the Set Variable activity but can be used with the Variable Compare and Collect Digits activities. Variables that contain file paths to .wav files will also secure the .wav file. System variables cannot be secured.



6.5.1 Add a Variable

To add a variable, perform the following steps:

1. Click  icon on the top right side of the Workflow Canvas page.
The **Variable Stack** configuration window is displayed.
2. Click **Add Variable**.
3. On the **Add Variable** page, enter the following details in the respective fields:
 - a. In the **Variable name** text field, enter a unique variable name.
 - b. From the **Type** dropdown, select the variable type from the following options: Boolean (True/False), Number, String, Object, Array.
 - c. In the **Default Value** text field, enter a default value, if applicable.
 - d. In the **Description** text field, enter a description if required.
 - e. Select the **Is Secure** toggle button to secure and mask the variable from the display.
4. Click **OK** to save the variable.

6.5.2 Delete a Variable

To delete a variable, perform the following steps:

1. Click  icon on the top right side of the Workflow Canvas page.
The **Variable Stack** configuration window is displayed.
2. In the **Search variables** text box, select a variable from the list or search for it by entering its name.
3. Click  icon to delete the variable.
4. Click **OK**.

6.5.3 Assigning Values to Variables

Assigning Literal Values

A literal value, such as a string, number, or object, can be assigned to a variable. For example, `myNumericVar = 6` or `myStringVar = '123'`.

When assigning a literal value to a variable, ensure that the variable option is set to Literal, as shown below. In this example, the string variable `myStringVar` is assigned the value 'hello'.

abc
123

myStringVar (String)
hello

×

To assign a variable to the literal value of another variable, enclose the variable in `{{ }}`, as shown below. In this example, the string variable `myStringVar` is assigned to the value of `myStringVar2`. If `myStringVar2` has the value of 'goodbye', after this assignment, the value of `myStringVar` will also be set to 'goodbye'. This notation can also be used in other activities whenever the literal value of the variable is used. It is important to note that this syntax only works for variables of type: `String`.

abc
123

myStringVar (String)
{{myStringVar2}}

×

Assigning Expressions

It is also possible to calculate expressions based on values of variables, throughputs, and outputs from activities that have previously run in a flow, and then assign the output of the expression to a variable. For example, `myNumericVar = myVar1 + myVar2` or `myStringVar = StringReplace(myStringVar2, 'abc', 'xyz')`.

To assign an expression to a variable, ensure that the variable option is set to Expression (fx), as shown below. In this example, the string variable `myStringVar` is assigned the value of the variables `str1`, `str2`, and `str3`. If the values are one, two, and three, respectively, then the value of `myStringVar` will be 'onetwothree' after this variable assignment completes.

fx

myStringVar (String)
StringBuilder(str1, str2, str3)

×

List of Actions

This chapter contains the following sections:

- [Media Actions](#)
- [User Management Actions](#)
- [Developer Actions](#)
- [Routing Actions](#)
- [MCX Actions](#)
- [Messaging Actions](#)
- [Integrations Actions](#)
- [Search Only Actions](#)

As an administrator, you can access the various actions from the **Workflow Canvas**.

The **Workflow Canvas** page displays the list of configured actions.

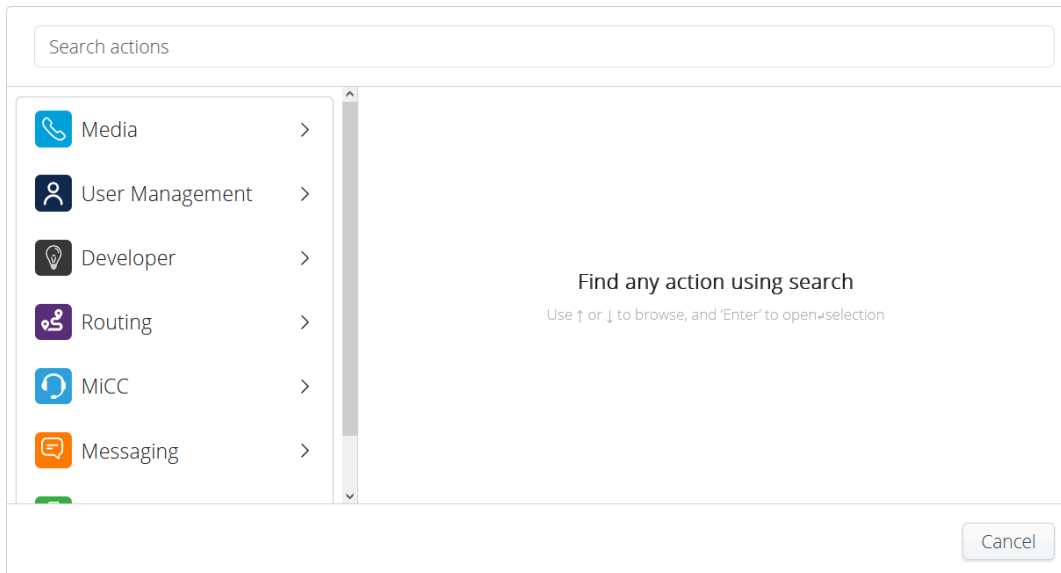


Figure 66: Search Actions

The action can be configured using the two views:

- **Classic View:**

By default, the **Classic** view is displayed. It allows you to configure all required input values for the activity.

- **Advanced View:**

Each activity also has an **Advanced** view, which displays all the input, throughput, and output values for the activity.





Click **Advanced** on the top right corner of the activity to access the **Advanced** view.








The **Advanced** view is intended for expert-level users. In some cases, it can configure input values for the activity that are not shown in the **Classic** view. For example, the Dial by Name activity searches for the user by first name by default. To change this to search by last name, select the **Advanced** view and modify the **SearchBy** input variable from *firstName* to *lastName*. This input value is only configurable from the **Advanced** view.

7.1 Media Actions

The following table describes the various actions supported under **Media** category in the **Workflow Canvas**:

Table 24: Media Actions

Action Name	Icon	Availability	License Type
Dial menu		Release	Essential
Greeting		Release	Essential
Dial by name		Release	Premier
Transfer		Release	Essential

Action Name	Icon	Availability	License Type
Make call		Release	Essential
Answer		Release	Essential
Hang up call		Release	Essential
Record audio message		Release	Premier
Get call tag		Release	Essential
DTMF tones		Beta	Essential
Tag call		Release	Essential

7.1.1 Dial menu

This activity offers a caller a menu of options, prompting the caller to select service options by entering a configured digit or speaking a response. When ASR is used, Mitel recommends asking the caller to say the requested response name instead of a numeric response to make it easier for ASR to recognize

the response. For example, "Press **1** or say **Service** to reach the service department". The workflow will branch to different activities depending on the caller's selection.

Table 25: Dial menu action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
AudioSource	The name of the prompt to be played if the option to "Use existing greeting" is selected.
TTSText	The text to be played to the caller if the option "Text to speech" is selected.
TTSVoice	The voice to be used to play the TTS text if the option "Text to speech" is selected.
MinDigits	If the caller is expected to enter digits, this is the minimum number of digits required to be entered. If no digits are expected, the value should be 0.
MaxDigits	If the caller is expected to enter digits, this is the maximum number of digits that can be entered. If no digits are expected, the value should be 0.
RepeatTimes	The number of times that the greeting will be repeated if the caller is expected to respond to the greeting and no response is received before the timeout.
TerminateOnAbandon	Indicates whether the workflow will exit if the caller hangs up.
TerminatingDigit	If the caller is expected to enter digits, entering this digit will return the digits entered, even if the timeout has not been reached, or the maximum number of allowed digits has not been entered.
EnterDigitTimeoutSeconds	If the caller is expected to respond to the greeting, this is the number of seconds the workflow will wait for the response (digits or ASR) before timing out.
UseASR	Indicates whether the caller can respond to the greeting with speech.

Parameters	Description
ASRLanguage	If UseASR is enabled, this parameter indicates the language to be used for ASR.
ResetRepeatCount	This indicates whether the RepeatTimes value will be reset when the action starts. It is useful if a GoTo action is used to point back to the Greeting action, so that the repeat count can be continued from the previous iteration and not reset.
ASRInitialSilenceDetection	<p>The numeric value representing the number of seconds to wait for the caller to begin speaking after the message plays if UseASR is enabled. If the value is greater than 0, the length of the message will automatically be added to this value so that the system will wait for the greeting to complete and then apply the ASRInitialSilenceDetection value.</p> <p>The value can range from 0-59 seconds. The default value is 30 seconds.</p>
ASRSilenceDetection	The numeric value representing the number of seconds of silence allowed before speech detection terminates if UseASR is enabled. The value can range from 0-59 seconds. The default value is 5 seconds.

Table 26: Dial menu action - Outputs

Parameters	Description
CollectedDigits	If a response was received from the caller, this parameter returns the digits entered by the caller.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	Returns the JSON formatted response from the activity.
TimesRepeated	Indicates the number of times that the greeting was repeated. A greeting will be repeated if a response is expected from the caller and is not received before the timeout.
CollectedSpeech	If an ASR response was received from the caller, this parameter returns the speech spoken by the caller.

Parameters	Description
BranchId	Returns "Tones" if digits or speech are collected from the caller. Returns "Failure" if no input is received from the caller or returns.
BranchResult	Returns the collected digits when BranchId returns "Tones".

7.1.2 Greeting

This activity allows a caller to provide input by entering a digit or speaking, similar to the **Dial menu** activity. It allows the workflow to collect input from the caller, and then process that input for cases that do not require a selection menu—for example, collecting speech from the caller and passing it to an AI Chatbot.

Table 27: Greeting action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
AudioSource	The name of the prompt to be played if the option to "Use existing greeting" is selected.
TTSText	The text to be played to the caller if the option "Text to speech" is selected.
TTSVoice	The voice to be used to play the TTS text if the option "Text to speech" is selected.
CallLeg	The call leg on which the message will be played. This can be Caller, Callee, or both.
MinDigits	This is the minimum number of digits required to be entered. If no digits are expected, the value should be 0.
MaxDigits	This is the maximum number of digits that can be entered. If no digits are expected, the value should be 0.
RepeatTimes	The number of times a greeting will be repeated, if the caller is expected to respond to the greeting and no response is received before the timeout.

Parameters	Description
TerminateOnAbandon	Indicates whether the workflow will exit if the caller hangs up.
TerminatingDigit	Entering this digit will return the digits entered, even if the timeout has not been reached or the maximum number of allowed digits has not been entered.
EnterDigitTimeoutSeconds	The number of seconds the workflow will wait for the response (digits or ASR) before timing out.
UseASR	Indicates whether the caller can respond to the greeting with speech.
ASRLanguage	If UseASR is set, this parameter indicates the language to be used for ASR.
ResetRepeatCount	This indicates whether the RepeatTimes value will be reset when the action starts. It is useful if a GoTo action is used to point back to the Greeting action so that the repeat count can be continued from the previous iteration and not reset.
ASRInitialSilenceDetection	<p>The numeric value representing the number of seconds to wait for the caller to begin speaking after the message plays if UseASR is enabled. If the value is greater than 0, the length of the message will automatically be added to this value so that the system will wait for the greeting to complete and then apply the ASRInitialSilenceDetection value.</p> <p>The value can range from 0-59 seconds. The default value is 30 seconds.</p>
ASRSilenceDetection	The numeric value representing the number of seconds of silence allowed before speech detection terminates if UseASR is enabled. The value can range from 0-59 seconds. The default value is 5 seconds.

Table 28: Greeting action - Outputs

Parameters	Description
CollectedDigits	This parameter returns the digits entered by the caller.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

Parameters	Description
Result	Returns the JSON formatted response from the activity.
TimesRepeated	Indicates the number of times that the greeting was repeated. A greeting will be repeated if a response is expected from the caller and is not received before the timeout.
CollectedSpeech	If an ASR response was received from the caller, this parameter returns the speech spoken by the caller.

7.1.3 Dial by name

This activity allows customers to find a specific employee in your company directory and then be automatically transferred to that person. Customers search the directory by entering two or more characters in the user's name through the customer's telephone keypad or by speaking the user's name. It is possible to exclude users such as the company CEO from the search results.

Table 29: Dial by name action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
AudioSource	The name of the prompt to be played if the option to "Use existing greeting" is selected.
TTSText	The text is to be played to the caller if the option "Text to speech" is selected.
MatchedAudioSource	The name of the prompt to be played when a user is found that matches the input criteria, when the option to "Use existing greeting" is selected.
MatchedTTSText	The text to be played to the caller when a user is found that matches the input criteria, when the option "Text to speech" is selected.
NoMatchAudioSource	The name of the prompt to be played when no user is found that matches the input criteria when the option to "Use existing greeting" is selected.
NoMatchTTSText	The text to be played to the caller when no user is found that matches the input criteria when the option "Text to speech" is selected.

Parameters	Description
TTSVoice	The voice used to play the TTS text if the option "Text to speech" is selected.
AcceptDigit	The digit expected to be entered by the caller to accept a transfer to the user.
SearchBy	Option to search by first or last name.
TerminateAfterTransfer	Indicates whether the workflow should exit after the caller is transferred to the target destination.
ExcludedUsers	A list of users that cannot be looked up by the activity.
UseASR	Indicates whether the caller can use ASR to return a response.
ASRLanguage	If UseASR is set, this parameter indicates the language to be used.
AcceptTerm	If UseASR is set, this parameter indicates the word to be detected using which the caller wishes to transfer to the destination.
ExitTerm	If UseASR is set, this parameter indicates the word to be detected using which the caller wishes to exit the activity.
SearchTerm	If UseASR is set, this parameter indicates the word to be detected using which the caller wishes to search for more users.
MaxRetry	The maximum number of search attempts before the activity exits.
ASRInitialSilenceDetection	<p>The numeric value representing the number of seconds to wait for the caller to begin speaking after the message plays if UseASR is enabled. If the value is greater than 0, the length of the message will automatically be added to this value so that the system will wait for the greeting to complete and then apply the <code>ASRInitialSilenceDetection</code> value.</p> <p>The value can range from 0-59 seconds. The default value is 30 seconds.</p>

Parameters	Description
ASRSilenceDetection	The numeric value representing the number of seconds of silence allowed before speech detection terminates if UseASR is enabled. The value can range from 0-59 seconds. The default value is 5 seconds.

Table 30: Dial by name action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

7.1.4 Transfer

This activity transfers an active call to the provided destination. If the call is being transferred to a MiCC queue, data can be sent to MiCC. See the [MiCC send data](#) activity for more information.

Table 31: Transfer action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
Destination	Number to which the call will be transferred.
TimeoutSeconds	Used for Supervised transfer, which is not currently supported.
TerminateOnAbandon	Indicates whether the workflow will exit if the caller hangs up.
CstaConnectionId	Additional information about the call to be transferred.
MiccbVarArray	If the MiCC Send data activity is used to identify variables to be sent to MiCC when the call is transferred, this input indicates the array of variables and values to be sent to MiCC.
SendToVoiceMailDirectly	This can be set to True to automatically transfer the call to the user's voice mailbox.
VMPilotNumberOverride	Number to be used for voicemail if the SendToVoiceMailDirectly option is set.

Parameters	Description
PbxType	If the value is "mivb" and the SendToVoiceMailDirectly option is set, it retrieves the voice mail destination number unless it is overridden by the VMPilotNumberOverride input value.

Table 32: Transfer action - Outputs

Parameters	Description
BranchResult	Indicates the branch to be executed based on the result of the transfer: "Success", "Failure", "Timeout" or "Abandon".
BranchId	Returns "Failure" or "Success".
Result	Returns the JSON formatted response from the activity.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

7.1.5 Make call

This activity makes an outbound call to the configured destination from another device. Passing in an alternate From Name to the Make Call activity can override the calling party name.

Table 33: Make call action - Inputs

Parameters	Description
FromName	Name to be presented for the calling party.Defaults to 'system'.
To	Number to be called.
CallingNumberOverride	Set to the value to be used in place of the actual calling number.

Table 34: Make call action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

Parameters	Description
CallId	Identifier of the call created.
State	State of the call created.
BranchId	Returns "Success" or "Failure"
BranchResult	Returns the call object.
CallRecord	Record with additional information about the outgoing call.
StatusCode	HTTP status code returned when requesting to initiate the call.
Body	Returns the call object.

7.1.6 Answer

This activity answers an inbound, ringing call. It is typically used with an Endpoint Ringing trigger.

Table 35: Answer action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
TerminateOnAbandon	Indicates whether the workflow will exit if the caller hangs up.

Table 36: Answer action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	Returns the JSON formatted response from the activity.

7.1.7 Hang up call

This activity ends an active call being handled by the workflow.

Table 37: Hang up call action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.

Table 38: Hang up call action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
StatusCode	HTTP status code returned when requesting to disconnect the call.
Body	Returns the JSON formatted response from the activity.

7.1.8 Record audio message

This activity lets the caller record a short audio message for the configured number of seconds. After the tone, the caller can record a message. The activity can be configured to hang up the call after the recording completes or allow the call to continue.

Table 39: Record audio message action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
MaxDurationSeconds	The maximum length of the recorded message.
HangupOnCompletion	Whether the call should be disconnected after the audio message is completed.
AlternateBeepMessage	To override the system beep tone, set this input value to another prompt defined in the system. Prompts can be recorded using the Greeting or Dial menu activity.

Table 40: Record audio message action - Outputs

Parameters	Description
RecordingId	Unique identifier of the created recording.
RecordingPath	Path to the recorded file.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
RecordingURLPath	URL for the recorded file.

7.1.9 Get call tag

This activity answers an inbound, ringing call. It is typically used with an Endpoint Ringing trigger.

Table 41: Get call tag action - Inputs

Parameters	Description
TerminateOnAbandon	Whether to exit the workflow if the caller hangs up.
TagId	Tag identifier to be retrieved; if left blank, all tags against the call will be returned.
CallId	Unique identifier for the active call.

Table 42: Get call tag action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Results	Returns the JSON payload of the tag(s) linked to the call ID.

7.1.10 DTMF tones

This activity sends DTMF tones to an active call. To override the default duration of the tone from 2 seconds, specify the duration after the tone in milliseconds. For example, "1234@4000" will play the tones 1, 2, 3, and 4 with a 4 second delay between each tone.

Table 43: DTMF tones action - Inputs

Parameters	Description
CallId	Unique identifier for the active call.
Tones	Digits to be sent as DTMF tones.
TerminateOnAbandon	Whether to exit the workflow if the caller hangs up.

Table 44: DTMF tones action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Body	Response to the request.
StatusCode	Number of tones sent.

7.1.11 Tag call

This activity adds a descriptive tag to an active call.

Table 45: Tag call action - Inputs

Parameters	Description
TagId	Identifier of the tag to be added.
TagValue	Value of the tag to be added.
TagObject	JSON list of tag identifiers and values if multiple tags are added.
CallId	Unique identifier for the active call.
TerminateOnAbandon	Whether to exit the workflow if the caller hangs up.

Table 46: Tag call action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	Response to the request.

7.2 User Management Actions

The following table describes the various actions supported under **User Management** category in the **Workflow Canvas**:

Table 47: User Management Actions

Action Name	Icon	Availability	License Type
Get user		Release	Essential
Search users		Release	Essential
User presence		Release	Essential

7.2.1 Get user

This activity searches for a CloudLink User based on the input parameter, which can be an email address, mobile, phone number, or userId.

Table 48: Get user action - Inputs

Parameters	Description
Identifier	Unique identifier of the user to be retrieved (email address, mobile number, phone number, or userId).
IncludeTags	Whether to return all tags associated with the user in the result.

Table 49: Get user action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	The user object, if found. otherwise, will be populated with an error message.

7.2.2 Search users

This activity retrieves one or more CloudLink users or PBX contacts matching a provided search term, such as a name or mobile number. The search term can be characters corresponding to a dial keypad or the actual name or number value. An array of matching users and contacts are returned.

Table 50: Search users action - Inputs

Parameters	Description
SearchTerm	Characters corresponding to a dial keypad if the SearchOption is set to "Keypad" or the actual characters to be searched for if the Search Option is set to "Standard".
SearchOption	Set to "Keypad" to search by digits or "Standard" to search for a match to the actual value.
SearchKey	Indicates the user attribute to be matched. The options are First Name, Last Name, Mobile, and Email.
Limit	The maximum number of users to return.
FilterDuplicateNumbers	Indicates whether to filter out users with the same number or not.

Parameters	Description
FilterKey	Filter parameter to be sent in the search query and formatted as an Odata filter parameter.

Table 51: Search users action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Returns an array with the list of users found.
Count	Returns the number of users found.

7.2.3 User presence

This activity retrieves or sets CloudLink presence for one or more selected users. Presence can be set to any of the predefined values, and an optional reason can be provided. The retrieved or set presence values are returned in an array.

Table 52: User presence action - Inputs

Parameters	Description
Operation	"Get" to retrieve presence, or "Set" to set presence for a user.
WorkMode	If set to true, setting the presence updates the agent Status and Reason. Otherwise, user presence is updated to the provided Status and Reason.
Users	List of users to set or retrieve presence for.
Status	Status to be set on the user.
Reason	Reason to be set on the user for the status update.





Table 53: User presence action - Outputs


Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	List of retrieved users with presence or list of updated users.

7.3 Developer Actions

The following table describes the various actions under **Developer** category supported in the **Workflow Canvas**:

Table 54: Developer Actions

Action Name	Icon	Availability	License Type
Condition		Release	Essential
Set variables		Release	Essential
Http request		Release	Essential
Log		Release	Essential

Action Name	Icon	Availability	License Type
Terminate		Release	Essential
For each		Release	Essential
Go to		Release	Essential
Break		Release	Essential
Delay		Release	Essential
Continue		Release	Essential
Bookmark		Release	Essential
Open session		Release	Essential

Action Name	Icon	Availability	License Type
JavaScript		Release	Premier
Cloudlink subscribe		Release	Essential
Cloudlink unsubscribe		Release	Essential
Parallel		Beta	Essential
Try-Catch		Release	Essential
While		Release	Essential

7.3.1 Condition

This activity configures conditions that branch the workflow, adds one or more branches, and groups multiple branches. This activity allows the workflow to take different actions depending on the customer's actions or the data received.

Inputs: expression to be evaluated to true or false.

7.3.2 Set variables

This activity sets the value of variables in the workflow. For more information, refer to the [Using Variables](#) topic.

7.3.3 Http request

This activity allows you to configure the HTTP activity to access RESTful Web APIs in your workflow. You can send requests using HTTP to web services to retrieve or update data. The HTTP activity encapsulates the header and body of the API request, making it easy to use within your workflow.

Table 55: Http request action - Inputs

Parameters	Description
Method	The HTTP action to execute (GET, PUT, POST, DELETE, HEAD, OPTIONS or CONNECT).
Url	The URL to send in the HTTP request.
Connection	Optional third-party connection object to be used for the HTTP request.
RetryPolicies	<p>An array of retry options for the HTTP request.</p> <p>Each array element is an object with the following syntax:</p> <pre>{ "ErrorStatusCodes": [429, 500], "RetryCount": 3, "RetryIntervalSec": 1 }</pre> <ul style="list-style-type: none"> • ErrorStatusCode: Defines the status codes matched against the http response to determine which retry policy to use. • RetryCount: Number of times the Http activity will retry the request before moving on. • RetryIntervalSec: The amount of time the Http activity will wait before retrying. The http activity will generate an exponential back-off interval based on the current retry count.

Table 56: Http request action - Throughputs

Parameters	Description
Headers	Array of header key/value pairs to be sent in the HTTP request.

Parameters	Description
Body	Contents of the HTTP request to be sent.

Table 57: Http request action - Outputs

Parameters	Description
RetryCount	Number of attempts to send the HTTP request.
StatusCode	HTTP status code returned from the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

7.3.4 Log

This activity adds log messages which provide additional information for debugging a flow.

Table 58: Log action - Inputs

Parameters	Description
Level	Indicates the log level of the message (Info, Warning, or Error).
Message	The message to be logged.

7.3.5 Terminate

This activity allows you to stop the workflow execution completely.

Table 59: Terminate action - Inputs

Parameters	Description
Reason	Optional reason to indicate why the workflow was terminated. This will be displayed in the workflow history.

7.3.6 For each

This activity allows the workflow to iterate through an array and execute activities on each item.

Table 60: For each action - Inputs

Parameters	Description
Collection	The array that will be iterated. The For each loop will execute for each item in the array.

7.3.7 Go to

This activity allows you to redirect the flow of a workflow. The workflow will continue executing the activities as indicated in this activity.

Table 61: Go to action - Inputs

Parameters	Description
Destination	Action to execute next in the workflow.

7.3.8 Break

This activity breaks the workflow from the current logical flow. This is commonly used with a **For Each** activity to stop iterating through the array and continue the workflow before all items are evaluated.

7.3.9 Delay

This activity allows you to delay or pause the workflow for a specified number of milliseconds (1000 milliseconds = 1 second).

Table 62: Delay action - Inputs

Parameters	Description
Delay	The number of milliseconds to delay.

7.3.10 Continue

This activity breaks from the current logical flow and continues at the next iteration of a loop. It is commonly used with a **For each** activity to stop the current iteration and continue with the next entry in the array.

7.3.11 Bookmark

This activity pauses the workflow for a specified time while waiting for an action. The workflow resumes if the action does not occur before the configured timeout value.

Table 63: Bookmark action - Inputs

Parameters	Description
Mode	'Pause' indicates that the workflow will wait for the configured amount of time for the action to occur. 'Save' pauses the workflow without a timeout.
Timeout	The length of time to wait for the action to occur. This is defined in days, hours, minutes, seconds, and milliseconds.
ContinueTimeoutOnResume	Set this value to True to keep the timer running even if the activity occurs that resumes the Bookmark activity. By default, the timer is canceled once the activity occurs.

Table 64: Bookmark action - Outputs

Parameters	Description
Resume	This value is set to True if the Bookmark activity resumed because the activity occurred. Otherwise, it is set to False if the Bookmark activity timed out.

7.3.12 Open session

This activity opens a new session for the workflow based on an identifier, such as a call ID or session ID. This is typically used in conjunction with a Bookmark activity. When the Open Session activity is executed, it will check if an existing session exists for the workflow running with the same identifier. If so, that session is resumed, and the existing Bookmark exits. Otherwise, a new session is created.

Table 65: Open session action - Inputs

Parameters	Description
SessionId	Unique identifier for the session.

7.3.13 JavaScript

This activity supports native JavaScript functions implemented directly in the JavaScript engine and does not require additional libraries. It can perform basic functions such as formatting or parsing data. The activity has access to all the workflow variables and can return a result to the workflow, along with additional data, when it is complete.

Table 66: JavaScript action - Inputs

Parameters	Description
Script	Valid JavaScript code to be executed.

Table 67: JavaScript action - Outputs

Parameters	Description
Data	Response from the JavaScript code.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)

7.3.14 Cloudfink subscribe

If the workflow requires Cloudfink notifications, the **Subscribe** activity can specify the topic, subject, and method, along with a filter for specific notification events. For example, to monitor the state of a call at an endpoint, the workflow can use this activity to receive notifications from platform-api-media for that specific endpoint and call. Refer to the [Cloudfink API documentation](#) for further details.


Note:

The notification events will be available in the system variable `notifications`.

Table 68: Cloudfink subscribe action - Inputs

Parameters	Description
Topic	These are used to define the Cloudfink events to be subscribed. Refer to the Cloudfink API documentation for details.
SubjectFilter	
Method	
PublicationFilter	

Table 69: Cloudlink subscribe action - Outputs

Parameters	Description
SubscriptionIds	Array of subscription identifiers created for the subscription. These can be used for Cloudlink unsubscribe activity.

7.3.15 Cloudlink unsubscribe

This activity cancels a subscription set up for CloudLink notifications through a Subscribe activity. The Subscribe activity returns the Subscription ID(s), which is passed to the Unsubscribe activity to stop receiving the requested CloudLink notifications.

Table 70: Cloudlink unsubscribe action - Inputs

Parameters	Description
SubscriptionIds	Array of subscription identifiers to be cancelled.

7.3.16 Parallel

This activity executes two or more branches in a workflow simultaneously. Each branch executes simultaneously, and the workflow waits for all branches to complete or one of the branches to complete, depending on the configuration of the Parallel activity.

Table 71: Parallel action - Inputs

Parameters	Description
Parallelism	'Wait on All' waits for all parallel activities to complete before moving forward in the workflow. 'Wait on One' waits for at least one activity to complete before moving forward in the workflow.

7.3.17 Try-Catch

This activity creates two branches: the "Try" branch will execute, and the workflow will continue with the "Catch" branch if an error is encountered. This allows you to add error handling to your script.

Table 72: Try-Catch action - Outputs

Parameters	Description
Exception	Object with information about the exception occurring if an exception is caught.

7.3.18 While

This activity causes a workflow to repeat activities until a condition is met. For example, while the variable count < 3, continue to repeat an activity and increment count. It can be executed in two different modes. "While" mode evaluates the condition first, and the loop is not executed if it is already met. "Do While" mode executes the loop and then evaluates the condition, so it is guaranteed to execute it at least once.

This activity is used with a Continue or Break activity to return to the While activity and evaluate the condition or break out of the While loop. This activity should be used carefully so that an infinite loop is not created.

Table 73: While action - Inputs

Parameters	Description
WhileType	Set to 'While' to evaluate the condition before the loop is executed or set to 'Do While' to evaluate the condition after the loop has executed. This ensures that the loop will always execute once.
Condition	Condition to be evaluated to determine whether the loop should continue.


Table 74: While action - Throughputs

Parameters	Description
Iteration	Indicates the number of times that the loop executed.

7.4 Routing Actions

The following table describes the various actions supported under **Routing** category in the **Workflow Canvas**:

Table 75: Routing Actions

Action Name	Icon	Availability	License Type
Schedule		Release	Essential

Action Name	Icon	Availability	License Type
Route interaction		Release	Essential
Emergency mode		Release	Essential
Phone number lookup		Release	Essential

7.4.1 Schedule

This activity allows you to set the working hours and holiday schedules for your organization.

If your account has Google integration, the Google Places API can automatically retrieve the working hours of the selected organization.

7.4.2 Route interaction

This activity routes a chat session to a specified queue.

Table 76: Route interaction action - Inputs

Parameters	Description
ChannelType	Type of interaction to be routed. Messaging option only supported.
Priority	Priority can be set to High, Normal, or Low.
Skills	Not currently used.
InteractionId	Identifier of the interaction to be routed.
SkillGroupId	Queue that the interaction is routed to.

Parameters	Description
DisplayTemplate	Not currently used.
Context	Record for the interaction with additional details.
TerminateWFO onSuccess	Set to True to cause the workflow to exit after the interaction is routed.

Table 77: Route interaction action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False)
Result	Returns the JSON formatted response from the activity.

7.4.3 Emergency mode

This activity turns on or off emergency mode for your account, allowing you to route interactions differently based on the configured setting.

Table 78: Emergency mode action - Inputs

Parameters	Description
Operation	Options are 'Check' to check the current status or 'Set' to set the status.
Status	Used when Operation is set to 'Set'. Options are 'Enabled' to set emergency mode or 'Disabled' to clear emergency mode.

Table 79: Emergency mode action - Outputs

Parameters	Description
BranchId	'Success' or 'Failure' depending on whether the operation succeeds or fails.
BranchResult	Current emergency status.

7.4.4 Phone number lookup

This activity allows you to look up a phone number to determine the approximate location attributes of the caller.

Table 80: Phone number lookup action - Inputs

Parameters	Description
PhoneNumber	The number to be looked up and parsed.

Table 81: Phone number lookup action - Outputs






Parameters	Description
PhoneNumberNational	The phone number parsed to national format, i.e. area code + number.
PhoneNumberE164	The phone number parsed to E.164 format, i.e. country code + area code + number.
Country	The readable name of the country that the number is from.
CountryCodeIso2	The ISO-2 abbreviation for the country that the number is from.
StateOrProvince	The state or province that the number is from.
City	The city that the number is from.
PhoneType	Indicates MOBILE for mobile number, LANDLINE for a connected extension or VOIP for a voice-over IP device.
Longitude	The longitude location of the device.
Latitude	The latitude location of the device.
Carrier	The mobile carrier, if the number is a mobile number.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

Parameters	Description
AreaCode	The area code for the number.
CountryCode	The numeric country code for the country that the number is from.

7.5 MCX Actions

The following table describes the various actions supported under **MCX** category in the **Workflow Canvas**:

Table 82: MCX Actions

Action Name	Icon	Availability	License Type
MCX get employee state		Release	Essential
MCX change employee state		Release	Essential
MCX get queue state		Release	Essential
MCX port state		Release	Essential
MCX route open media		Release	Essential

Action Name	Icon	Availability	License Type
MCX send data		Release	Essential
MCX callback		Release	Essential

7.5.1 MCX get employee state

This activity retrieves the current aggregate state for an MCX employee and an optional reason. The state can be any of the following values: Unknown, Available, Busy, Do Not Disturb, Away, or Offline.

Table 83: MCX get employee state action - Inputs

Parameters	Description
EmployeeId	Identifier of the MCX employee.

Table 84: MCX get employee state action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON formatted response to the request.
State	State of the employee (Unknown, Available, Busy, Do Not Disturb, Away or Offline).
Reason	Optional reason associated with the state.

7.5.2 MCX change employee state

This activity sets the aggregate state for an MCX employee with an optional reason. The state can be any of the following values: Unknown, Available, Busy, Do Not Disturb, Away, or Offline.

Table 85: MCX change employee state action - Inputs

Parameters	Description
EmployeeId	Identifier of the MCX employee.
Reason	Options include Contact Center Work Timer, No DND Reason Code, No Make Busy Code, Overloaded, Resize Timer and System Make Busy Code.
DeviceId	MCX Device ID.
HotDeskBaseExtension	Employee extension.
HotDeskPin	Employee extension PIN.
HotDeskAgentReporting	Employee extension reporting number.
IsCloudlinkAuthenticated	Indicates whether the employee is authenticated in Cloudlink.
IsHotDeskExternal	Indicates whether the hot desk number is an external number.
State	State to set on the employee (Available, Busy, Do Not Disturb, Away, or Offline).
AvailableReason	Optional reason for the state.

Table 86: MCX change employee state action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

Parameters	Description
Body	JSON formatted response to the request.

7.5.3 MCX get queue state

This activity retrieves the current real-time statistics for a MCX queue, including the number of available agents, the number of idle agents, the number of waiting conversations, and whether the queue is in a Do Not Disturb state.

Table 87: MCX get queue state action - Inputs

Parameters	Description
QueueId	Identifier of the MCX queue.

Table 88: MCX get queue state action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON formatted response to the request.
IsInDoNotDisturb	Indicates whether the queue is in Do not Disturb state.
AgentsAvailable	Number of agents available to serve the queue.
WaitingConversations	Number of conversations waiting in the queue.
AgentsIdle	Number of agents able to serve the queue but idle.

7.5.4 MCX port state

This activity sets or retrieves the current emergency setting of a MCX hunt group, extension, or media server. If the device is in emergency mode, alternate routing choices can be used in MCX, such as when there is an extreme weather situation.

Table 89: MCX port state action - Inputs

Parameters	Description
Operation	Options are 'Get' to retrieve the current device state or 'Set' to set the current device state.
DeviceTypeController	Indicates the type of device, i.e., Hunt Groups, Media Servers or Extensions.
DeviceDialable	Indicates the unique identifier for the device.

Table 90: MCX port state action - Throughputs

Parameters	Description
Status	Options are 'Normal' to set the device to normal state or 'Emergency' to set emergency mode. If the Operation is 'Set', this indicates the current device state.

Table 91: MCX port state action - Outputs

Parameters	Description
BranchId	'Success' if the operation succeeds, or 'Failure' if it fails.
BranchResult	Same value as Status

7.5.5 MCX route open media

This activity sends an Open Media request to the MCX system to be routed to a MCX agent. The request can be any data object, such as a social media message.

Table 92: MCX route open media action - Inputs

Parameters	Description
RequestBody	Formatted JSON object to send to MCX. Refer to the activity for options that can be specified when sending the open media request to MCX.

Table 93: MCX route open media action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.

7.5.6 MCX send data

This activity sends selected variables to MCX when a call is transferred. It allows the user to select which of the workflow variables will be sent to MCX, and then when a call is transferred, the selected variables are sent along with the call.

Table 94: MCX send data action - Inputs

Parameters	Description
VariableList	List of the names of variables defined in the workflow that will be sent to MCX with the transferred call.

Table 95: MCX send data action - Outputs

Parameters	Description
MiccbVarArray	Formatted array of name and value pairs indicating the selected variable names and the associated values.

7.5.7 MCX callback

This activity adds a callback to the MCX system, which can be routed to a MCX agent. The customer's name and number can be specified, as well as the callback destination number. It is required to specify the MCX media server that will handle the callback and the subroutine in MCX that will handle the callback routing.

Table 96: MCX callback action - Inputs

Parameters	Description
CustomerName	Name of the customer to be called.

Parameters	Description
CustomerNumber	Number of the customer.
CallRecord	JSON object with additional information about the call, such as ani and dnis.
DestinationNumber	Number to be dialed for the callback.
OtherCallbackProperties	Additional properties to be sent in the callback request, formatted as JSON.
MediaServer	MCX media server that will handle the callback.
CallbackSubroutine	Subroutine defined in MCX to process and route the callback.
VariableList	List of the names of variables defined in the workflow that will be associated with the callback request. If the variable does not exist in MCX, it will be created.








Table 97: MCX callback action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.



7.6 Messaging Actions

The following table describes the various actions supported under **Messaging** category in the **Workflow Canvas**:

Table 98: Messaging Actions

Action Name	Icon	Availability	License Type
Create chat stream		Release	Essential
Send chat message		Release	Essential
Ask		Release	Essential
Add participant		Release	Essential
Tag chat		Release	Essential
Delete chat		Release	Essential
Create chat transcript		Release	Essential

Action Name	Icon	Availability	License Type
Create or update attachment		Release	Essential
Get chat tag		Release	Essential
Get chats with tag		Release	Essential
Delete chat tag		Release	Essential
Delete participant		Release	Essential
Get participants		Release	Essential
Add endpoint participant		Release	Essential

Action Name	Icon	Availability	License Type
Send command		Release	Essential
Send whisper		Release	Essential

7.6.1 Create chat stream

This activity creates a new chat stream for a chat conversation.

Table 99: Create chat stream action - Inputs

Parameters	Description
Name	Name of the chat stream to be created.
Visibility	Options are 'Public' or 'Private', to restrict visibility to others.
DefaultParticipant	Identifier of a participant to be added to the chat stream after it is created.
DefaultParticipantAsGuest	If a DefaultParticipant is specified, indicates whether the default participant will be added as a guest or regular participant.
Hidden	Indicate 'true' to hide the chat stream.
GenerateSystemMessages	If set to 'true', system messages will be generated in the conversation.

Table 100: Create chat stream action - Outputs

Parameters	Description
ConversationId	Unique identifier for the created chat stream.

Parameters	Description
Result	Returns the JSON formatted response from the activity.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.6.2 Send chat message

This activity sends a chat message to all users in an existing chat conversation or creates a new conversation to send a direct message to selected users. The content type of the message is commonly text or HTML, but other advanced options are available, including Action Card, Location, Missed Call, Voicemail, or System Message.

Table 101: Send chat message action - Inputs

Parameters	Description
Body	Content of the message, if it is of type 'Text'.
BodyObject	JSON formatted content of the message for messages not of type 'Text'.
ContentType	Indicates the content type of the message to be sent. Options are 'Text', 'Html', 'Action Card', 'Location', or 'Missed Call', 'Voicemail', and 'System Message'.
Receivers	If a ConversationId is not specified, specifies a list of receivers to which the message will be sent. A new conversation is created in this case.
From	Identifier of the user sending the message, when a ConversationId is specified. The user must be an existing member of the conversation.

Table 102: Send chat message action - Throughputs

Parameters	Description
ConversationId	Unique identifier for the conversation, if it currently exists. If a direct message is sent, it returns the unique identifier for the created conversation.

Table 103: Send chat message action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Returns the JSON formatted response from the activity.

7.6.3 Ask

This activity sends a question to an existing chat conversation and returns the answer received. A timeout value can be configured to complete the activity if an answer is not received within the timeout period (default is 60 seconds).

Table 104: Ask action - Inputs

Parameters	Description
ConversationId	Unique identifier for the conversation.
BodyObject	JSON formatted content of the message for messages not of type 'Text'.
TimeoutSeconds	Amount of time in seconds to wait for a response to be received.
ContentType	Indicates the content type of the message to be sent. Options are 'Text', 'Html', 'Whisper', 'Location' or 'Action Card'.
Body	Content of the message, if it is of type 'Text'.

Table 105: Ask action - Outputs

Parameters	Description
Result	Returns the JSON formatted response from the activity.
Answer	Response received.
BranchId	Returns 'Responses' if responses are received or 'Timeout' if no response is received before the timeout period.

Parameters	Description
BranchResult	Response received.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.6.4 Add participant

This activity adds one or more participants to an existing chat conversation. The users can be added as guests or full participants. If the participant is not added as a guest, it is validated as a CloudLink user or client.

Table 106: Add participant action - Inputs

Parameters	Description
ConversationId	Unique identifier for the conversation.
Status	Options are 'INVITED' to invite the participants to the conversation, or 'ACCEPTED' to add the participants to the conversation automatically.
Participants	Array of unique identifiers for the participants to be added to the conversation.
AddAsGuest	Indicates whether the participants are added as guest members to the conversation.

Table 107: Add participant action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Returns an array of the JSON formatted responses for each participant.

7.6.5 Tag chat

This activity adds tags to an existing conversation. Tags can be used to categorize chat conversations.

Table 108: Tag chat action - Inputs

Parameters	Description
ConversationId	Unique identifier for the conversation.
Tags	Array of tag id and value pairs, if multiple tags are added.
TagId	Identifier of the tag, if a single tag is added.
TagValue	Value of the tag, if a single tag is added.

Table 109: Tag chat action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Returns the JSON formatted response from the activity.

7.6.6 Delete chat

This activity deletes an existing chat conversation. Messages will no longer be accepted for the chat conversation.

Table 110: Delete chat action - Inputs

Parameters	Description
ConversationId	Unique identifier for the conversation.

Table 111: Delete chat action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.

Parameters	Description
StatusCode	HTTP response code for the request.

7.6.7 Create chat transcript

This activity retrieves the transcription of a specific chat for a set duration.

Table 112: Create chat transcript action - Inputs

Parameters	Description
ConversationId	A drop-down list of streams in which the workflow caller is a participant.
Duration	The length of time that the transcript will contain (today, 7 days, 30 days, beginning)

Table 113: Create chat transcript action - Outputs

Parameters	Description
FormattedHtml	The transcription of the chat in HTML format.
IsSuccess	Indicates whether the activity was completed successfully (value = True) or not (value = False).
CompanyName	The name of the CloudLink Account.

7.6.8 Create or update attachment

This activity creates a new attachment or updates an existing attachment for an existing conversation. The content of the attachment can be sent in Base 64 encoded format, or a URL can be provided from which the attachment will be retrieved. The attachment can be up to 10MB.

Table 114: Create or update attachment action - Inputs

Parameters	Description
ConversationId	A drop-down list of streams that the caller of the workflow is a participant in.

Parameters	Description
ContentType	The HTTP Content-Type value for the type of the attachment, i.e. image/jpeg, application/pdf, etc.
AttachmentId	For an existing attachment, indicate the ID for the attachment.
AttachmentName	The display name for the attachment.
Base64EncodedBuffer	If the attachment content is provided, this variable holds the content in Base 64 format.
AttachmentFetchUrl	If the attachment content is not provided in the Base64EncodedBuffer variable, the location of the attachment content.
AttachmentFetchHeaders	If AttachmentFetchUrl is used, any headers required to access the attachment are added here.
UserId	To send the message as a guest, enter the user Id of the user. If the workflow is already running under a user who is a member of the conversation, this input is not required.

Table 115: Create or update attachment action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response from adding a message with the attachment if successful, or the failure response if the attachment fails to be added.

7.6.9 Get chat tag

This activity retrieve a tag from a chat conversation.

Table 116: Get chat tag action - Inputs

Parameters	Description
ConversationId	Chat conversation ID.

Parameters	Description
TagId	Name of the tag to be retrieved.

Table 117: Get chat tag action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Value of the retrieved tag.

7.6.10 Get chats with tag

This activity retrieves all chat conversations with a specific tag ID or value associated with the conversation.

Table 118: Get chats with tag action - Inputs

Parameters	Description
TagId	Identifier of the tag.
TagValue	Value of the tag.

Table 119: Get chats with tag action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	List of the returned chat conversations.
StatusCode	Number of chat conversations retrieved.

7.6.11 Delete chat tag

This activity deletes a tag from a chat conversation.

Table 120: Delete chat tag action - Inputs

Parameters	Description
ConversationId	Chat conversation ID.
TagId	Identifier of the tag to be removed (if a single tag)
TagIds	Array of tag identifiers to be removed (if multiple tags)

Table 121: Delete chat tag action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the delete request.

7.6.12 Delete participant

This activity remove a participant from a chat conversation.

Table 122: Delete participant action - Inputs

Parameters	Description
ConversationId	Chat conversation Id.
Participants	Array of participant IDs to be removed from the conversation.

Table 123: Delete participant action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.6.13 Get participants

This activity retrieves all participants for a chat conversation.

Table 124: Get participants action - Inputs

Parameters	Description
ConversationId	Chat conversation Id.

Table 125: Get participants action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response with the list of participants.

7.6.14 Add endpoint participant

This activity adds a workflow instance to a chat conversation so that the workflow can receive notifications for chat messages.

Table 126: Add endpoint participant action - Inputs

Parameters	Description
ConversationId	Chat conversation Id.
UserId	User ID of the user to be added to the conversation.
Endpoint	Workflow ID to be added to the conversation.

Table 127: Add endpoint participant action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the request.

7.6.15 Send command

This activity sends a custom command to a chat conversation.

Table 128: Send command action - Inputs

Parameters	Description
Arguments	Arguments for the command to be sent.
Command	Command to be sent.
ConversationId	Chat conversation Id.

Table 129: Send command action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the request.

7.6.16 Send whisper

This activity sends a whisper message to a chat participant, that is only visible to that participant.

Table 130: Send whisper action - Inputs

Parameters	Description
ConversationId	Chat conversation Id.
UserId	User that will receive the message.
Message	Message to be sent.

Table 131: Send whisper action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the request.

7.7 Integrations Actions





This topic describes the various actions supported under **Integrations** category in the **Workflow Canvas**.

7.7.1 JIRA Cloud Actions

The following table describes the various actions supported under **JIRA Cloud** category in the **Workflow Canvas**:

For the JIRA Cloud Integration procedure, refer to the [Jira Cloud Integration \(Premier\)](#) on page 264 section.

Table 132: JIRA Cloud Actions

Action Name	Icon	Availability	License Type
Jira add issue comment		Beta	Premier
Jira create issue		Beta	Premier
Jira get issue		Beta	Premier
Jira update issue		Beta	Premier

7.7.1.1 Jira add issue comment

This activity adds a comment to an existing issue in Jira.

Table 133: Jira add issue comment action - Inputs

Parameters	Description
Connection	Jira integration connections.
Issued	Issue Identifier.
Comment	The comment you wish to add to an existing ticket.

Table 134: Jira add issue comment action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.1.2 Jira create issue

This activity is used to create a new issue in Jira.

Table 135: Jira create issue action - Inputs

Parameters	Description
Connection	Jira integration connections.
Issue Type	EPIC / User Story / Task / Bug
ProjectKey	Project key to where the issue will be created.
Summary	Title of the issue.
Description	Description of the issue.

Parameters	Description
CustomFields	Additional settings for Jira custom fields, formatted as JSON.
ReporterId	Identifier of the user reporting the issue.
AssigneeId	Identifier of the user to whom the issue will be assigned.

Table 136: Jira create issue action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Id	Unique identifier assigned for the issue.
Key	Project key assigned for the issue.

7.7.1.3 Jira get issue

This activity returns all the information related to an issue.

Table 137: Jira get issue action - Inputs

Parameters	Description
Connection	Jira integration connections.
IssueId	Issue Identifier of the issue to be returned.

Table 138: Jira get issue action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity.

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.1.4 Jira update issue

This activity is used to modify an existing issue in Jira.

Table 139: Jira update issue action - Inputs

Parameters	Description
Connection	Jira integration connections.
Issueld	Issue Identifier.
Summary	Title of the issue.
Description	Description of the issue.

Table 140: Jira update issue action - Outputs








Parameters	Description
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).








7.7.2 Microsoft Graph Actions

The following table describes the various actions supported under **Microsoft Graph** category in the **Workflow Canvas**:

For the Microsoft Graph Integration procedure, refer to the [Microsoft Graph Integration \(Premier\)](#) on page 268 section.

Table 141: Microsoft Graph Actions

Action Name	Icon	Availability	License Type
Microsoft Graph validate user email		Release	Essential
Office 365 Outlook - Send email		Release	Essential
Excel add, update or delete table rows		Beta	Premier
Excel get table rows		Beta	Premier
Teams find user		Beta	Essential
Teams presence		Beta	Essential
Teams get team channels		Beta	Essential

Action Name	Icon	Availability	License Type
Teams get team details		Beta	Essential
Teams send adaptive card		Beta	Essential
Get calendar availability		Beta	Essential
Set calendar event		Beta	Essential
Send message to MS Teams channel		Beta	Premier
OneDrive - File upload		Release	Premier
OneDrive - Get file contents		Beta	Premier

7.7.2.1 Microsoft Graph validate user email

This activity looks up an email address on a configured Microsoft Graph API connection to validate whether the email exists.

Table 142: Microsoft Graph validate user email action - Inputs

Parameters	Description
ConnectionId	Connection to Microsoft Graph API.
Email	Email address to be validated.

Table 143: Microsoft Graph validate user email action - Outputs

Parameters	Description
StatusCode	The HTTP response code returned from the lookup. For example, 200 indicates that the email address was validated, and 404 indicates that the email address could not be found.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the response from the Microsoft Graph API for the validation request. If the lookup is successful, the response will include other email addresses for the user.

7.7.2.2 Office 365 Outlook – Send email

This activity sends an email through Office 365 to selected email addresses. A connection must first be configured to Microsoft Office 365 – Outlook from Workflow to use this activity. The email content, recipients (including CC and BCC recipients), and email attachments can be specified.

Table 144: Office 365 Outlook – Send email action - Inputs

Parameters	Description
Connection	Drop-down list of Microsoft Office 365 connections.
To	Array of email addresses to which the email will be sent.
Cc	Array of email addresses to be sent a copy of the email.
Bcc	Array of email addresses to be sent a blind copy of the email.
Subject	Subject of the email.

Parameters	Description
ContentType	Options are 'html' or 'text'.
Body	Body of the email.
Importance	Options are 'normal', 'low' and 'high' to indicate the importance of the email when it is delivered.
ReplyTo	Array of email addresses to be used when receivers reply to the email.
RequestDeliveryReceipt	Whether to request a delivery receipt when the email is delivered to the receiver.
RequestReadReceipt	Whether to request a read receipt when the email is read by the receiver.
UserId	User associated with the Microsoft Office 365 connection, if required.
Sender	Email address of the sender.
From	Normally the same as the Sender value, unless this is a shared mailbox, and then the owner of the mailbox can be specified.
Attachments	Array of file names for attachments to be sent.
AttachmentUrls	Array of urls to be sent as attachments. The attachment will be downloaded from the url and attached to the email.

Table 145: Office 365 Outlook – Send email action - Outputs

Parameters	Description
ResponseBody	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.3 Excel add, update or delete table rows

This activity add a new row of data to an Excel table. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 146: Excel add, update or delete table rows action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
UserId	The query will be run under the selected userID. The user should be the owner of the Excel file; otherwise, it will fail to update.
Workbook	Name of the.xlsx file.
Sheet	The tab name on the worksheet to which the values must be modified.
Table	The defined name of the table.
Operation	Add, Update or Delete.
Values	For the Add or Update operations, this is a list of data values to be added or updated to the selected worksheet cells.
Cells	Array of cells to be updated.
DeleteRowIndex	For the Delete operation, indicates the row to be deleted. The index is 0-based and related to the start of the table.

Table 147: Excel add, update or delete table rows action - Outputs

Parameters	Description
UpdatedRowCount	The number of rows updated.

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.4 Excel get table rows

This activity retrieves the rows of data based on the filter provided. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 148: Excel get table rows action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
UserId	The query will be run under the selected userID. The user should be the owner of the Excel file; otherwise, it will fail to update.
Workbook	Name of the xlsx file.
Sheet	The tab name on the worksheet to which the values must be modified.
Table	The defined name of the table.
Filter	Filter criteria to be applied for the query. Refer to the Graph API for information on the filter format.

Table 149: Excel get table rows action - Outputs

Parameters	Description
Rows	The rows of data that are returned from the query.
RowCount	Number of rows returned.

7.7.2.5 Teams find user

This activity retrieves a list of Azure Users based on the filter provided. Please note that the filter will return the top 500 if left blank. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 150: Teams find user action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
Filter	An expression to filter the returned list, for example, <code>displayName eq 'FullName'</code> .

Table 151: Teams find user action - Outputs

Parameters	Description
User	Returns the User requested to be retrieved.
Body	Returns the JSON formatted response from the activity.
Status	Returns the HTTP response code for the request. For example, 200 indicates a successful response.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.6 Teams presence

For an existing Microsoft Graph API connection, this activity can be used to set, retrieve or clear the presence of a user defined on the Microsoft Azure system. The user's ID is an input value to the activity, along with an action value indicating Set, Get or Clear presence. If Set presence is requested, the formatted body of the request must also be provided to the activity to specify the presence value, as shown below:

```
{
  "sessionId": "22553876-f5ab-4529-bffb-cfe50aa89f87",
  "availability": "Available",
  "activity": "Available",
  "expirationDuration": "PT1H"
}
```

Refer to the **Microsoft Graph REST API documentation** for details on the options for setting presence.

A connection must be created to Microsoft Graph API to use this activity. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 152: Teams presence action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
Id	Unique identifier of the user.
Action	Options are <ul style="list-style-type: none"> • Set - set the presence for a user. • Clear - clear the presence for a user. • Get - retrieve the presence for a user. • Set User Presence - set a user defined presence for a user, such as a custom presence reason. • Clear User Presence - clear a user defined presence for a user.
Payload	Formatted body of the request if 'Set' or 'Set User Presence' are selected.

Table 153: Teams presence action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Headers	Array of HTTP headers from the response.
Body	Returns the JSON formatted response from the activity.

7.7.2.7 Teams get team channels

This activity retrieves all the channels for a team. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 154: Teams get team channels action - Inputs

Parameters	Description
TeamId	The team's ID of the directory where the channel exists or will be created. Refer to the Retrieve the Teams ID section for instructions on how to obtain the Teams ID.
Connection	Drop-down list of Graph connections.

Table 155: Teams get team channels action - Outputs

Parameters	Description
Body	Returns the JSON-formatted response from the activity.
StatusCode	Returns the HTTP response code for the request. For example, 200 indicates a successful response.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.8 Teams get team details

This activity retrieves the details of a team. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 156: Teams get team details action - Inputs

Parameters	Description
TeamId	The team's ID of the directory where the channel exists or will be created. Refer to the Retrieve the Teams ID section for instructions on how to obtain the Teams ID.
Connection	Drop-down list of Graph connections.

Table 157: Teams get team details action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity.

Parameters	Description
StatusCode	Returns the HTTP response code for the request. For example, 200 indicates a successful response.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Name	Returns the name of the team.
MemberCount	Number of members in the team.

7.7.2.9 Teams send adaptive card

This activity sends adaptive card info to a team, such as "Typing" or "Suggestion". For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 158: Teams send adaptive card action - Inputs

Parameters	Description
Text	Contains the message text to be sent to the conversation, when the Type is not set to Message.
ConversationId	Chat conversation Id.
Type	Options are: <ul style="list-style-type: none"> • Message: An adaptive card is provided with message information. • Typing: A message indicating that the person is typing is sent. • Message Reaction: A message reaction, such as an emoji, is sent. • End of Conversation: A message indicating that the conversation has ended is sent. • Suggestion: A message with pre-defined options, such as suggested responses, is sent. • Trace: The message provides debugging or troubleshooting information.
AdaptiveCard	This input provides the contents of the adaptive card to be sent to Microsoft Teams. It is used when the Type is set to 'Message'. For assistance designing an adaptive card, refer to the Microsoft Teams documentation.

Parameters	Description
Connection	Drop-down list of Graph connections.

Table 159: Teams send adaptive card action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity which includes alternate time slots and the user's current schedule.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.10 Get calendar availability

This activity looks up a specific date and time period on the user's calendar in the Microsoft Azure system to determine if the user is available. If the user is not available, alternate times can be returned. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 160: Get calendar availability action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
UserEmail	Email address of the user whose calendar is retrieved.
Duration	Duration of time in minutes to check for a free time slot.
StartDateTime	Start date and time in UTC format to be used to check for a free time slot.
RetrievePermissions	If set to true, an object indicating the calendar permissions is returned.
NumberOfOpeningsToReturn	If the requested date and time is not available, indicates the number of alternate time slots to return.

Table 161: Get calendar availability action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity which includes alternate time slots and the user's current schedule.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Schedule	Returns the user's calendar.
CalendarPermissions	If 'RetrievePermissions' is set to true, returns an object with the calendar permissions.

7.7.2.11 Set calendar event

This activity sets a new event in the user's calendar on the Microsoft Azure system for a specific date and time period. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 162: Set calendar event action - Inputs

Parameters	Description
Connection	Drop-down list of Graph connections.
UserEmail	Email address of the user whose calendar is retrieved.
StartDateTime	Start date and time in UTC format to be used to check for a free time slot.
Location	Location for the meeting, such as a meeting room or Online.
AttendeeEmails	Array of emails for the users to be invited to the meeting.
Duration	Duration of the meeting in minutes.
EventName	Descriptive name for the meeting.

Parameters	Description
OnlineMeeting	Set to 'true' to indicate that the meeting is online.

Table 163: Set calendar event action - Outputs

Parameters	Description
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.2.12 Send message to MS Teams channel

This activity allows you to send a message to an MS Teams Channel and supports sending action cards, HTML, etc. If you provide a channel name that doesn't exist, it will create a new channel; otherwise, it will send the message to the existing channel. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 164: Send message to MS Teams channel action - Inputs

Parameters	Description
ChannelName	Name of the MS Teams Channel to which the message has to be sent.
ChannelMessage	A message to be displayed inside the channel. MS Teams can handle multiple message types, including action cards.
ChannelDescription	A description of the channel. This is used when a new channel is created.
TeamsId	The team's ID of the directory where the channel exists or will be created. Refer to the Retrieve the Teams ID section for instructions on how to obtain the Teams ID.
UserConnection	Graph connection for the user sending the message into the channel.
AppConnection	An application-based Graph connection that is used to search for whether the channel name provided exists.

Table 165: Send message to MS Teams channel action - Outputs

Parameters	Description
StatusCode	Return the HTTP code based on the result. For example, 200 indicates success.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON payload of the response returned by the API.

7.7.2.13 OneDrive - File upload

This activity allows you to upload a file to a location on OneDrive. Please note that the connection must be configured as OAuth2 Authorization Code. For Azure permissions, refer to the [Microsoft Azure Integration](#) section.

Table 166: OneDrive File upload action - Inputs

Parameters	Description
FolderName	The name of the folder inside the OneDrive account to which the file will be uploaded.
FileURLLocation	The URL of the globally accessible file that you want to copy into your OneDrive.
Connection	A drop-down of Graph connections. Please note that only OAuth2 Auth Code connections will work, as the file can only be uploaded to the owner of the connection.
Filename	The file name that will be saved inside the OneDrive account.
IsShareable	True or False. Setting it to true will allow anyone in the organization to have access to the file.

Table 167: OneDrive File upload action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

Parameters	Description
Results	JSON payload of the action performed.
DownloadURL	A link from which the user can access the file.
ShareableURL	A link from which anyone in the organization can access the file.

7.7.2.14 OneDrive - Get file contents

This activity provides the ability to retrieve the contents of a file as a binary stream.

Table 168: OneDrive - Get file contents action - Inputs

Parameters	Description
Connection	A drop down of Graph connections, please note only OAuth2 Auth Code connections will work as the file can only be downloaded by the owner of the connection.
FileList	A drop down of all the files under that One Drive account. The activity will search recursively through all of the folders and display any .pdf, .docx or .txt files.

Table 169: OneDrive - Get file contents action - Outputs




Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Results	JSON payload of the action performed.

7.7.3 Salesforce Actions

The following table describes the various actions supported under **Salesforce** category in the **Workflow Canvas**:

For the Salesforce Integration procedure, refer to the [Salesforce Integration \(Premier\)](#) on page 296 section.

Table 170: Salesforce Actions

Action Name	Icon	Availability	License Type
Salesforce create, update or delete record		Beta	Premier
Salesforce get record field values		Beta	Premier
Salesforce query SOQL		Beta	Premier

7.7.3.1 Salesforce create, update or delete record

This activity updates, creates, or deletes a record based on a Salesforce Object (SObject).

Table 171: Salesforce create, update or delete record action - Inputs

Parameters	Description
Connection	A Salesforce connection.
Method	Options are: <ul style="list-style-type: none"> • Create - create a new record in Salesforce • Update - update an existing record in Salesforce • Delete - delete a record from Salesforce

Parameters	Description
RecordBody	<p>A JSON body of the record that is being affected.</p> <p>For example,</p> <pre>{ "Name": "Updated Product Name", "ProductCode": "UPD001", "IsActive": true, "Description": "This is an updated description of the product.", "Family": "Updated Product Family" }</pre>
SObject	A drop-down of a list of Salesforce objects
SObjectId	The identifier of a record, which is only required when the method type is Update or Delete.

Table 172: Salesforce create, update or delete record action - Outputs

Parameters	Description
Body	Returns a JSON response being returned by the execution of the operation
StatusCode	Return the HTTP code based on the result. For example, 200 is a success
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.3.2 Salesforce get record field values

This activity retrieves a record based on a Salesforce Object (SObject) selected.

Table 173: Salesforce get record field values action - Inputs

Parameters	Description
Connection	A Salesforce connection.
SObject	Drop down of a list of Salesforce objects.
SObjectID	The identifier of a record.
Fields	An array of Salesforce column names to be returned.

Table 174: Salesforce get record field values action - Outputs

Parameters	Description
Body	Returns a JSON response returned by the execution of the operation
StatusCode	Return the HTTP code based on the result. For example, 200 indicates success.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.3.3 Salesforce query SOQL

This activity performs an operation based on the SOQL provided, which is a SQL-like language.

Table 175: Salesforce query SOQL action - Inputs

Parameters	Description
Connection	A Salesforce connection.
SOQL	<p>A query field.</p> <p>For example, SELECT Id, Name, ProductCode, Family, and IsActive FROM Product2.</p> <p>Refer to the Salesforce documentation for more details on the format of an SOQL query.</p>

Table 176: Salesforce query SOQL action - Outputs




Parameters	Description
Body	Returns a JSON response being returned by the execution of the operation.
StatusCode	Return the HTTP code based on the result. For example, 200 indicates success.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Records	Array of retrieved records.

7.7.4 CM.com Actions

The following table describes the various actions supported under **CM.com** category in the **Workflow Canvas**:

For the CM.com integration procedure, refer to the [CM.com Integration \(Premier\)](#) on page 232 section.

Table 177: CM.com Actions

Action Name	Icon	Availability	License Type
CM.com send message		Release	Premier
CM.com PUT API		Release	Premier
CM.com POST API		Release	Premier

7.7.4.1 CM.com send message

This activity sends an SMS message from a CM.com account to a specified number. A connection must be configured to CM.com to use this activity.

Table 178: CM.com send message action - Inputs

Parameters	Description
Connection	Drop-down list of CM.com connections.
Message	Text message to be sent.
To	Number to which the message is sent.
From	CM.com user to send the message from.
Channel	Select the channel to be used for the message. Options are 'WhatsApp Business', 'SMS' and 'Facebook Messenger'.
MediaURL	Optional URL of media attachment for the message.
MediaURLName	Name to be associated with the media attachment.

Table 179: CM.com send message action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.

7.7.4.2 CM.com PUT API

This activity is used as a helper activity to call into the CM.com API. This should only be used by users who understand the CM.com REST API documentation.

7.7.4.3 CM.com POST API





This activity is used as a helper activity to call into the CM.com API. This should only be used by users who understand the CM.com REST API documentation.

7.7.5 Github Actions

The following table describes the various actions supported under **Github** category in the **Workflow Canvas**:

For the GitHub Integration procedure, refer to the [GitHub Integration \(Essential\)](#) on page 239 section.

Table 180: Github Actions

Action Name	Icon	Availability	License Type
Github create issue		Beta	Essential
Github get issue		Beta	Essential
Github update issue		Beta	Essential
Github add issue comment		Beta	Essential

7.7.5.1 Github create issue

This activity provides the ability to create a Github issue in a specific repository.

Table 181: Github create issue action - Inputs

Parameters	Description
Title	The name of the issue.
Description	An explanation of the issue.
Owner	Name of the resource.
Repository	The name of the repository that the issue resides in, please do not include .git in the name.
Connection	A drop-down list of Github connections configured via the Github Integration.

Table 182: Github create issue action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON payload of the action performed.
StatusCode	HTTP Status code returned.

7.7.5.2 Github get issue

This activity retrieves a specific issue from the selected Github repository.

Table 183: Github get issue action - Inputs

Parameters	Description
Connection	A drop-down list of Github connections configured via the Github Integration.
Repository	The name of the repository that the issue resides in, please do not include .git in the name.

Parameters	Description
Owner	Name of the resource.
IssueNumber	The number of the issues that are to be retrieved.

Table 184: Github get issue action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON payload of the action performed.
StatusCode	HTTP Status code returned.

7.7.5.3 Github update issue

This activity provides the ability to update an existing Github issue in a specific repository.

Table 185: Github update issue action - Inputs

Parameters	Description
Connection	A drop down of Github connections configured via the Github Integration.
Repository	The name of the repository that the issue resides in, please do not include .git in the name.
Owner	Name of the resource.
IssueNumber	The number of the issues that are to be retrieved.
Title	The name of the issue
Description	An explanation of the issue.

Parameters	Description
State	The state of the issue. The options are Open or Closed.
Labels	A list of strings that will be classified as labels.
Assignees	A list of people who are assigned the issue.

Table 186: Github update issue action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON payload of the action performed.
StatusCode	Return the HTTP code based on the result.

7.7.5.4 Github add issue comment

This activity adds a comment to a Github issue.

Table 187: Github add issue comment action - Inputs

Parameters	Description
Owner	Name of the resource.
Repository	The name of the repository that the issue resides in, without .git in the name.
Connection	A drop-down list of Github connections configured via the Github Integration.
Comment	Comment for the Github issue.
IssueNumber	The number of the issue that is to be updated.





Table 188: Github add issue comment action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	JSON payload of the action performed.
StatusCode	HTTP Status code returned.

7.7.6 MiTeams Meeting Actions

The following table describes the various actions supported under **MiTeams Meeting** category in the **Workflow Canvas**:

Table 189: MiTeams Meeting Actions

Action Name	Icon	Availability	License Type
Create meeting		Beta	Essential
Delete meeting		Beta	Essential
Get meeting		Beta	Essential
Update meeting		Beta	Essential

7.7.6.1 Create meeting

This activity creates a MiTeam meeting.

Table 190: Create meeting action - Inputs

Parameters	Description
Name	Name of the meeting.
Participants	List of participant CloudLink user IDs or guest participants.
InviteOnly	Set to True to send an invitation to the user but don't automatically join the user to the meeting.
LobbyPolicy	Whether to immediately add users to the meeting or have them wait in the lobby until they are accepted.

Table 191: Create meeting action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the request.
AccessCode	Access code for the meeting.
StatusCode	Response HTTP status code.
Body	HTTP response to the request.
ConferenceId	Conference identifier for the meeting.

7.7.6.2 Get meeting

This activity retrieves a meeting by the conference identifier or meeting access code.

Table 192: Get meeting action - Inputs

Parameters	Description
MeetingIdOrAccessCode	Meeting identifier or access code of the meeting to be retrieved.

Table 193: Get meeting action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	HTTP response to the request.
AccessCode	Meeting access code.
MeetingId	Meeting identifier.

7.7.6.3 Delete meeting

This activity deletes a MiTeam meeting by the meeting Id or access code.

Table 194: Delete meeting action - Inputs

Parameters	Description
MeetingIdOrAccessCode	Meeting identifier or access code of the meeting to be deleted.

Table 195: Delete meeting action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
StatusCode	Returns the HTTP code based on the result. For example, 200 indicates success.
Body	HTTP response to the request.

7.7.6.4 Update meeting

This activity updates an existing meeting with a new name or updated participants.

Table 196: Update meeting action - Inputs

Parameters	Description
Name	Name of the meeting.
Participants	List of participant CloudLink user IDs or guest participants.
MeetingIdOrAccessCode	Meeting identifier or access code of the meeting to be retrieved.


Table 197: Update meeting action - Outputs

Parameters	Description
Result	Response to the request.
StatusCode	Response HTTP status code.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	HTTP response to the request.


7.7.7 Other Integration Actions

The following table describes the other integration actions in the **Workflow Canvas**:

Table 198: Search Only Actions

Action Name	Parent Integration	Icon	Availability	License Type
Twilio send message	Twilio		Release	Premier

Action Name	Parent Integration	Icon	Availability	License Type
Twilio - Get message	Twilio		Release	Premier
Twilio Get Content Template API	Twilio		Release	Premier
ChatGPT	OpenAI		Beta	Premier
OpenAI Assistants	OpenAI		Beta	Premier
OpenAI Text-to-speech	OpenAI		Beta	Essential
OpenAI Speech-to-text	OpenAI		Release	Premier
Google Drive - File upload	Google Drive		Beta	Essential

Action Name	Parent Integration	Icon	Availability	License Type
Google Drive - Get file contents	Google Drive		Beta	Essential
Google Gemini	Google Gemini		Beta	Premier
Anthropic Claude	Anthropic		Beta	Premier

7.7.7.1 Twilio send message

This activity sends an SMS message from a Twilio account to a specified number. To use this activity, a connection must be configured to Twilio.

Table 199: Twilio send message action - Inputs

Parameters	Description
Connection	Drop-down list of Twilio connections.
From	Twilio user to send the message from.
To	Number to which the message is sent.
Body	Message to be sent.
AccountSid	Twilio security identifier for the account.
MediaUrl	Optional URL of media attachment for the message.

Table 200: Twilio send message action - Outputs

Parameters	Description
ReponseBody	Returns the JSON formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.7.2 Twilio – Get message

This activity fetches a message from your Twilio account using the Programmable Messaging API. To use this activity, a connection must be configured to Twilio.

Table 201: Twilio – Get message action - Inputs

Parameters	Description
Connection	Drop-down list of Twilio connections.
AccountSid	Twilio security identifier for the account.
MessageSid	Twilio security identifier for the message.

Table 202: Twilio – Get message action - Outputs

Parameters	Description
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Message	Returns the response message from the activity.

7.7.7.3 Twilio Get Content Template API

This activity retrieves content templates defined in Twilio. To use this activity, a connection must be configured to Twilio.

Table 203: Twilio Get Content Template API action - Inputs

Parameters	Description
Route	Specify the information to be retrieved from Twilio, such as Content. Refer to the Twilio Content API for all possible Route values.
Connection	Drop-down list of Twilio connections.
PageRecords	Set to True to retrieve one page of records. Additional requests to the Twilio Get Content Template API activity will be required to retrieve the additional records. Set to False to retrieve all records in a single request.
ResponsePropertyName	The name of the template property that will be returned in the response. Refer to the Twilio Content API documentation for additional information.
FieldPropertyNames	The names of the properties that will be displayed in the template. Refer to the Twilio Content API documentation for additional information.

Table 204: Twilio Get Content Template API action - Outputs

Parameters	Description
Body	Returns the JSON-formatted response from the activity.
StatusCode	HTTP response code for the request.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.7.4 ChatGPT

For an existing OpenAI connection, this activity can send a message to the ChatGPT chatbot and receive a response. The message to be sent and the model to interpret the message are input values to the activity. For details on the supported model options, refer to the OpenAI documentation. The activity returns the response as a string; the full response is in JSON format. To use this activity, a connection must be configured to OpenAI.

Table 205: ChatGPT action - Inputs

Parameters	Description
Prompt	The message to be sent to ChatGPT.
Connection	Drop-down list of OpenAI connections.
Model	The OpenAI model to be used Prompt: Information to be sent to ChatGPT.

Table 206: ChatGPT action - Outputs

Parameters	Description
Response	Response string from ChatGPT.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Output	JSON response from ChatGPT.

7.7.7.5 OpenAI Assistants

This activity enables the workflow to use the OpenAI Assistants API to build an artificial intelligence (AI) assistant into your workflow. It allows instructions to be sent to the AI, as well as requests which the AI assistant responds to. For more details, refer to the OpenAI Assistants API documentation. To use this activity, a connection must be configured to OpenAI.

Table 207: OpenAI Assistants action - Inputs

Parameters	Description
Connection	Drop-down list of OpenAI connections.
Assistant	Unique identifier for the OpenAI assistant.
Message	Information to be sent to the OpenAI assistant.
Instructions	Training and background information to be sent to the OpenAI assistant.

Table 208: OpenAI Assistants action - Throughputs

Parameters	Description
ThreadId	Identifier of the thread assigned to handle the request. This can be passed into subsequent requests to use the same instance of OpenAI.

Table 209: OpenAI Assistants action - Outputs

Parameters	Description
Response	Response string from OpenAI.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Output	JSON response from OpenAI.

7.7.7.6 OpenAI Text-to-speech

This activity plays a message using one of the OpenAI text-to-speech voices. If the speech is collected from the caller, it will wait for the user to respond and return the response from the user in the activity response output variable..

Table 210: OpenAI Text-to-speech action - Inputs

Parameters	Description
Connection	Drop-down list of OpenAI connections.
Voice	Select one of the supported voices to be used for the speech.
Message	Specify the message to be converted to speech.
CollectSpeech	Set to True to wait for a response from the caller and return that value in the response. Set to False to just play the message.

Table 211: OpenAI Text-to-speech action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
CollectedSpeech	If CollectSpeech is set to True, this field represents the response received from the caller.

7.7.7.7 OpenAI Speech-to-text

This activity transcribes an audio file and has the ability to translate the audio file into OpenAI-supported languages as described in [OpenAI](#) documentation.

Table 212: OpenAI Speech-to-text action - Inputs

Parameters	Description
AudioFormat	Select the audio format as wav or mp3, depending on the file format of the speech sample.
URL	Specify the URL to the file that will store the text output.
Connection	Drop-down list of OpenAI connections.
Translate	Select English to translate the speech to English or None to maintain the current language.
Model	Select the OpenAI model to be used for the translation.

Table 213: OpenAI Speech-to-text action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
STT	Object representing the output of the speech-to-text translation.

7.7.7.8 Google Drive - File upload

This activity is used to upload a file from a shared URL to a location specified on a Google Drive.

Table 214: Google Drive - File upload action - Inputs

Parameters	Description
FolderName	Name of the folder where the file will be uploaded, or empty to update to the root folder.
FileURLLocation	URL representing the location of the file.
Connection	Drop-down list of Google connections.
FileName	Name of the file when it is stored on the Google Drive.

Table 215: Google Drive - File upload action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.7.7.9 Google Drive - Get file contents

This activity is used to return the contents of any selected files that exist in a Google Drive account. The supported files are .pdf, .txt or .docx.

Table 216: Google Drive - Get file contents action - Inputs

Parameters	Description
Connection	Drop-down list of Google connections.
Selected files	Array of selected file names. Files can be .pdf, .txt or .docx files.

Table 217: Google Drive - Get file contents action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
FileList	Array of the contents of the retrieved files.

7.7.7.10 Google Gemini

This activity enables the workflow to use the Google Gemini artificial intelligence chatbot. It provides an easy-to-use interface where instructions can be sent to the chatbot, along with a message. The output of the activity is the response from the chatbot. To use this activity, a connection must be configured to Google Gemini.

Table 218: Google Gemini action - Inputs

Parameters	Description
Connection	Drop-down list of Google Gemini connections.
Model	Google Gemini model to be used.
Message	Information to be sent to Google Gemini.
Instructions	Training and background information to be sent to Google Gemini.

Table 219: Google Gemini action - Outputs

Parameters	Description
Response	Response string from Google Gemini.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Output	JSON response from Google Gemini.

7.7.7.11 Anthropic Claude

This activity enables the workflow to use the next-generation artificial intelligence (AI) assistant Claude from Anthropic. It provides an easy-to-use interface where instructions can be sent to the AI, along with a message. The output of the activity is the response from the AI assistant. To use this activity, a connection must be configured to Anthropic.

Table 220: Anthropic Claude action - Inputs

Parameters	Description
Connection	Drop-down list of Anthropic connections.
Model	Anthropic model to be used.
Message	Information to be sent to Anthropic.
Instructions	Training and background information to be sent to Anthropic.

Table 221: Anthropic Claude action - Outputs


Parameters	Description
Response	Response string from Anthropic.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Output	JSON response from Anthropic.




7.8 Search Only Actions

The following table describes the various uncategorized actions in the **Workflow Canvas** and available in **Search** only option:

Table 222: Search Only Actions

Action Name	Parent Integration	Icon	Availability	License Type
Detect sentiment	AWS		Beta	Premier
Translate	AWS		Beta	Premier
Send message - 69xx phone	General		Release	Essential
Persisted variables	General		Release	Essential
Weather	General		Beta	Premier
Publish notification	CloudLink		Release	Essential
Slack send	Slack		Beta	Essential

Action Name	Parent Integration	Icon	Availability	License Type
MiCollab phone number parser	MiCollab		Beta	Essential
Create contact	Contacts		Release	Essential
Delete contact	Contacts		Release	Essential
Tag user	Contacts		Release	Essential
Add record MiVB form	MiVB		Beta	Premier
Delete record MiVB form	MiVB		Beta	Premier
Get MiVB form data	MiVB		Beta	Premier

Action Name	Parent Integration	Icon	Availability	License Type
Update record MiVB form	MiVB		Beta	Premier
AI CloudLink Chat Bot	AI		Beta	Premier
AI Voice Bot	AI		Beta	Premier

7.8.1 Detect sentiment

This activity determines whether the tone of a text string is positive, negative, or neutral. Supports English and Spanish languages.

Table 223: Detect sentiment action - Inputs

Parameters	Description
Language	Supports English or Spanish.
Text	Text information to be analyzed.

Table 224: Detect sentiment action - Outputs

Parameters	Description
Sentiment	Indicates POSITIVE, NEGATIVE or NEUTRAL after analyzing the input string
Result	Object with a numeric score after evaluating the text string for the following values: Positive, Negative, Neutral, Mixed.

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.8.2 Translate

This activity translates a text string from a source language to a target language. Supports the following languages: English, Arabic, French, German, Chinese, Portuguese, and Spanish.

Table 225: Translate action - Inputs

Parameters	Description
Text	Text string to be translated
SourceLanguage	Language to be translated from
TargetLanguage	Language to be translated to

Table 226: Translate action - Outputs

Parameters	Description
Result	Object containing the Source Language Code, Target Language Code, and Translated Text.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Translated Text	Translation of the input Text string to the Target Language.

7.8.3 Send message - 69xx phone

This activity sends message to a 69xx series phone, currently only supported by MiVoice Business.

Table 227: Send message - 69xx phone action - Inputs

Parameters	Description
TopTitle	Additional title to be displayed on the phone.

Parameters	Description
Title	Main message title to be displayed on the phone.
Users	List of users to send the message to. Messages can be sent to users, dialable numbers, or groups.
DialableNumbers	List of numbers to send the message to.
Groups	List of groups to send the message to.
Options	Options that can be set for the message, in JSON format.

Table 228: Send message - 69xx phone action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Response to the request.
StatusCode	HTTP status code returned from the request.

7.8.4 Persisted variables

This activity retrieves, sets or deletes a persisted variable. This can be used to store a variable that should exist across multiple Workflow sessions.

Table 229: Persisted variables action - Inputs

Parameters	Description
Method	The HTTP action to execute (GET, POST, or DELETE).

Table 230: Persisted variables action - Throughputs

Parameters	Description
Records	If the Method is set to POST, represents the values to be set for the variable. If the Method is set to GET, this represents the values retrieved for the variable.

Table 231: Persisted variables action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
StatusCode	HTTP status code returned from the request.

7.8.5 Weather

This activity looks up details about the weather for a location based on the city/country or latitude/longitude.

Table 232: Weather action - Inputs

Parameters	Description
City	Name of the city
CountryCode	Optional two-character country code based on the ISO 3166-1 alpha-2 standard.
Lat	Latitude coordinates of the location.
Lon	Longitude coordinates of the location.
Unit	Indicates whether the temperature is returned in Metric or Imperial format.

Table 233: Weather action - Outputs

Parameters	Description
Result	Returns details about the weather, including a description of the weather, temperature, humidity percentage, and barometric pressure.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Temp	Returns the temperature in the units requested.

7.8.6 Publish notification

This activity publishes a notification event on the CloudLink platform. Internal or 3rd party subscribers will receive the event if they are subscribed to the event topic. Notify any internal or 3rd party subscriber application will receive the event based on the subscription settings.

Table 234: Publish notification action - Inputs

Parameters	Description
Topic	CloudLink Topic
Method	There are three types of notifications being sent: <ul style="list-style-type: none"> • Post - New entry • Put - Update • Delete - Removal of an item
Subject	This identifies what is being updated
Content	The object of the content.
PrincipleFilter	A list of conditions that will be applied to the notification; only when true will the application receive the event.

Table 235: Publish notification action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Results	The body that is being sent in the notification.

7.8.7 Slack send

This activity sends a message to a Slack channel. An incoming webhook must be configured. To do this, create a Slack App and get your Slack Administrator to approve it.

Table 236: Slack send action - Throughputs

Parameters	Description
Channel	The name of the Slack channel that the message will be sent to.
Username	The Slack username that will be sending the message to the channel
Message	The text to be sent to the Channel
URL	The Slack Webhook URL generated when creating the incoming webhook

Table 237: Slack send action - Outputs

Parameters	Description
Results	The payload returned from the post of the URL to Slack.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.8.8 MiCollab phone number parser

This activity validates a phone number.

Table 238: MiCollab phone number parser action - Inputs

Parameters	Description
PhoneNumber	Number to be validated

Table 239: MiCollab phone number parser action - Outputs

Parameters	Description
Result	Returns the number in National, International, and E.164 format.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

7.8.9 Create contact

This activity creates a new contact in CloudLink.

Table 240: Create contact action - Inputs

Parameters	Description
ContactId	Unique identifier for the contact.
Name	Name of the contact.
Email	Email address of the contact.
ContactType	Technical Support, Main Contact, On-Site, Maintainer or Guest.

Table 241: Create contact action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the HTTP request.

7.8.10 Delete contact

This activity deletes a contact from CloudLink.

Table 242: Delete contact action - Inputs

Parameters	Description
ContactId	Unique identifier for the contact.

Table 243: Delete contact action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the HTTP request.

7.8.11 Tag user

This activity sets descriptive tags on a CloudLink user.

Table 244: Tag user action - Inputs

Parameters	Description
Identifier	Unique identifier for the contact.
Tags	JSON list of tag name and values.

Table 245: Tag user action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Response to the HTTP request.

7.8.12 Add record MiVB Form

This activity is used to add a record in MiVoice Business form.

Table 246: Add record MiVB Form action - Inputs

Parameters	Description
FormURL	Selects the MiVB view to be used to add a record in MiVB form. The options are Hunt Group, Caller Based Routing, Ring Group Members, Ring Group, and Telephone Directory.
RequestBody	Specify the body of the request in JSON format or a variable. Refer to the MiVB API documentation for format details.

Table 247: Add record MiVB Form action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.
StatusCode	HTTP status code returned for the request.

7.8.13 Delete record MiVB Form

This activity is used to delete a record from MiVoice Business.

Table 248: Delete record MiVB Form action - Inputs

Parameters	Description
FormURL	Selects the MiVB view to be used to delete the record in MiVB form. The options are Hunt Group, Caller Based Routing, Ring Group Members, Ring Group, and Telephone Directory.
Identifier	Specify the unique identifier for the record to be deleted.

Table 249: Delete record MiVB Form action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON-formatted response from the activity.
StatusCode	HTTP status code returned for the request.

7.8.14 Get MiVB form data

This activity is used to retrieve data from a MiVoice Business system.

Table 250: Get MiVB form data action - Inputs

Parameters	Description
FormURL	Selects the MiVB view to be used to retrieve data. The options are Hunt Group, Caller Based Routing, Ring Group Members, Ring Group, and Telephone Directory.
SearchValue	The value for which to search. To find a value within another value, the syntax %value% can be used. For example, searching for %sales% will match with Sales Group , Sales and External Sales.
SearchField	The numeric field number that will be searched.
ComponentId	Displays the list of MiVB systems registered. Select the MiVB for the request. If this is not specified, the active MiVB will be selected.

Table 251: Get MiVB form data action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON formatted response from the activity.

Parameters	Description
StatusCode	HTTP status code returned for the request.

7.8.15 Update record MiVB Form

This activity is used to update a record in MiVoice Business system.

Table 252: Update record MiVB Form action - Inputs

Parameters	Description
FormURL	Selects the MiVB view to be used to update the record in MiVB form. The options are Hunt Group, Caller Based Routing, Ring Group Members, Ring Group, and Telephone Directory.
RequestBody	Specify the body of the request in JSON format. For additional details, refer to the MiVB documentation.
Identifier	Specify the name of the item to be updated in MiVB. Refer to the MiVB documentation for additional details.
ComponentId	Displays the list of MiVB systems registered. Select the MiVB for the request. If this is not specified, the active MiVB will be selected.

Table 253: Update record MiVB Form action - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Body	Returns the JSON-formatted response from the activity.
StatusCode	HTTP status code returned for the request.

7.8.16 AI CloudLink Chat Bot

This activity uses a Large Language Model AI bot within a CloudLink chat flow.

Table 254: AI CloudLink Chat Bot action - Inputs

Parameters	Description
ConnectionType	The AI providers to be used are Open AI, Google Gemini, or Anthropic.
Connection	An existing connection to the AI provider.
Model	The model of the AI provider to be used.
Instructions	Plain language instructions to be provided to the AI bot.
BreakoutKeywords	Keywords provided to the AI bot.
ErrorMessage	Error message to be sent if the request to the AI bot fails.
Message	Message to be sent to the AI bot.

Table 255: AI CloudLink Chat Bot action - Outputs

Parameters	Description
Response	Response from the AI bot to the Message input parameter.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Full result sent from the AI bot

7.8.17 AI Voice Bot

This activity uses a generative AI bot within a voice flow.

Table 256: AI Voice Bot action - Inputs

Parameters	Description
ConnectionType	The AI providers to be used are Open AI, Google Gemini, or Anthropic.

Parameters	Description
Connection	An existing connection to the AI provider.
Model	The model of the AI provider to be used.
Greeting	Greeting message to be played to the caller.
Voice	Voice to be used to play the greeting message.
Language	Language to be used to play the greeting message.
Instructions	Plain language instructions to be provided to the AI bot.
BreakoutKeywords	Keywords to provide to the AI bot.
ErrorMessage	Error message to be played if the request to the AI bot fails.

Table 257: AI Voice Bot action - Outputs

Parameters	Description
Response	Response from the AI bot to the caller's input.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Full result sent from the AI bot.

List of Triggers

This chapter contains the following sections:

- [Messaging Triggers](#)
- [Media Triggers](#)
- [User Management Triggers](#)
- [Integrations Triggers](#)
- [Developer Triggers](#)
- [Search Only Triggers](#)

As an administrator, you can access the various triggers from the **Workflow Canvas**.

The **Workflow Canvas** page displays the list of configured triggers.

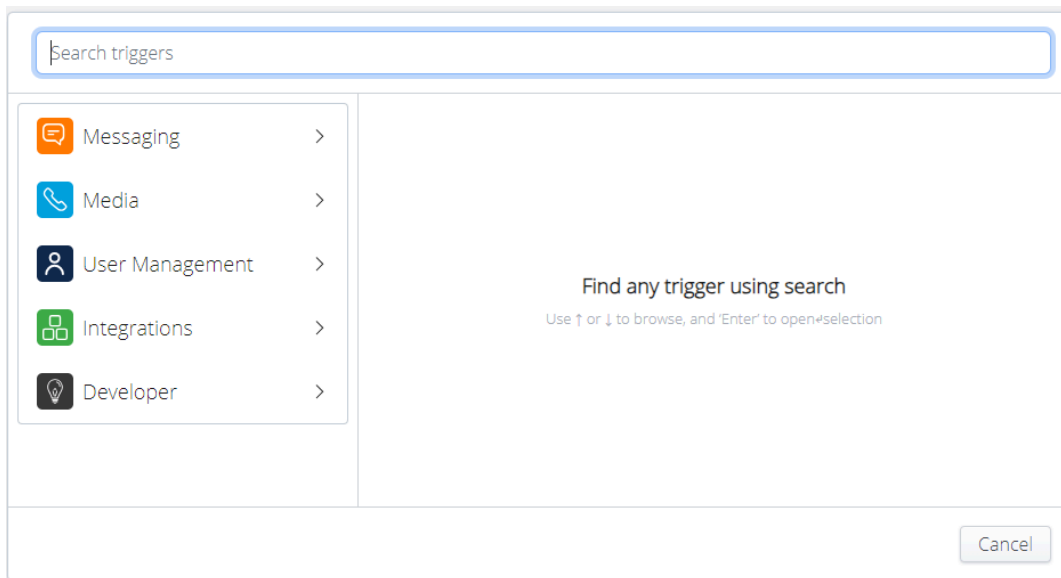


Figure 67: Search Triggers

8.1 Messaging Triggers

The following table describes the various triggers supported under **Messaging** category in the **Workflow Canvas**:

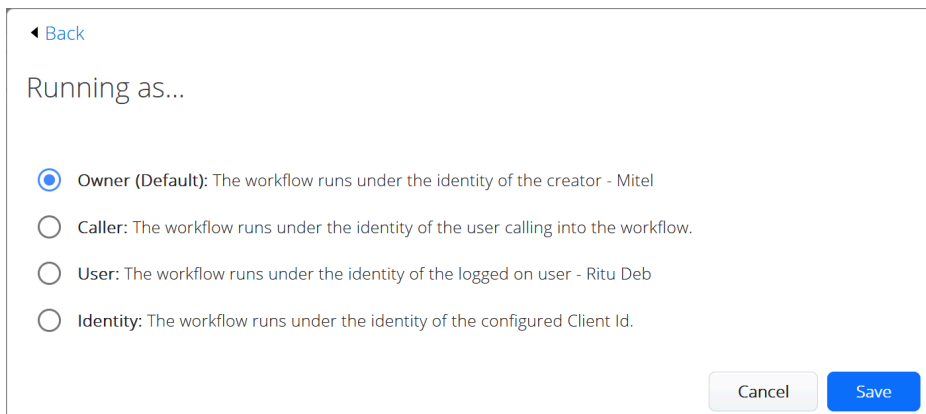
Table 258: Messaging Triggers

Trigger Name	Icon	Availability	License Type
Chat message received		Release	Essential
Chat created		Release	Essential
CC Messenger		Release	Essential

8.1.1 Chat message received

This trigger allows notification when a chat message is received in an existing chat stream or direct message. The owner of the workflow needs to be a participant in the conversation.

Under the **Actions** menu, select the **Set run flow as** to view the owner of the flow as shown in the image below.



◀ Back

Running as...

☒ Owner (Default): The workflow runs under the identity of the creator - Mitel

☐ Caller: The workflow runs under the identity of the user calling into the workflow.

☐ User: The workflow runs under the identity of the logged on user - Ritu Deb

☐ Identity: The workflow runs under the identity of the configured Client Id.

Cancel Save

Figure 68: Running as

Table 259: Chat message received trigger - Inputs

Parameters	Description
ConversationId	The id of the stream or direct message. * Indicates all streams or direct messages in which the flow owner participates.
From	The name of the participant that sent the message.
IncludeMe	Flag indicating whether to notify when the workflow owner sends a message or when other users send a message.

Table 260: Chat message received trigger - Outputs

Parameters	Description
Body	Message text received.
Result	JSON response with additional information about the message.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

8.1.2 Chat created

This trigger receives a notification any time there is a new chat stream. Note that the workflow and the stream's creator must be the same user to receive the notification. This notification is not available for direct messages.

Table 261: Chat created trigger - Outputs

Parameters	Description
Conversation	JSON object with the properties of the chat conversation.
ConversationId	Unique identifier of the chat conversation.

8.1.3 CC Messenger

This trigger receives a notification when a customer starts a web chat using the embedded chat configured dialog.

Table 262: CC Messenger trigger - Inputs

Parameters	Description
ExistingConversationId	Unique identifier for the conversation.

Table 263: CC Messenger trigger - Outputs

Parameters	Description
SessionId	Unique identifier for the session.
TrackingData	Array of information about the conversation.
TargetGroup	MiCC queue associated with the conversation.
CustomerId	Unique identifier for the customer sending the message.
BotResult	JSON formatted object with information about the chat when using the chat bot.
MetaData	JSON formatted object with additional information about the conversation.
ConversationId	Unique conversation identifier.
InteractionId	MiCC interaction identifier.
SkillGroupId	Not currently used.
Name	Name of the customer sending the message.
Email	Email address of the customer sending the message.
CustomerInformation	JSON formatted object with additional information about the customer.

8.2 Media Triggers

The following table describes the various triggers supported under **Media** category in the **Workflow Canvas**:

Table 264: Media Triggers

Trigger Name	Icon	Availability	License Type
Endpoint ringing		Release	Essential
Extension state changed		Release	Essential

8.2.1 Endpoint ringing

This trigger allows you to receive notifications on incoming SIP trunk calls. It is used with Voice Assist or when a workflow needs to act on an incoming call.

Table 265: Endpoint ringing trigger - Inputs

Parameters	Description
Endpoint	List of trunk numbers to select from, or you can add a new trunk number from the drop-down list.

Table 266: Endpoint ringing trigger - Outputs

Parameters	Description
CallId	Unique identifier for the call.
TimeZoneId	ISO identifier for the time zone from which the caller is calling.
CallRecord	JSON formatted object with additional information about the call.

Parameters	Description
InteractionId	Unique interaction identifier.
CallerName	Name of the caller.
CallerNumber	Number of the caller.
DialedNumber	Number originally dialed by the caller.
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).

8.2.2 Extension state changed

This trigger allows the workflow to receive a notification for an extension when it changes to a different state (Ringing, Connected, *)

Table 267: Extension state changed trigger - Inputs

Parameters	Description
Extension	The extension number for which you wish to receive notifications.
State	The phone state that you want to trigger for. The options are Any, Ringing, or Connected.





Table 268: Extension state changed trigger - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	JSON formatted response.
CallId	Unique identifier for the call.

8.3 User Management Triggers

The following table describes the various triggers supported under **User Management** category in the **Workflow Canvas**:

Table 269: User Management Triggers

Trigger Name	Icon	Availability	License Type
Presence changed		Release	Essential
User changed		Release	Essential
User deactivation/ deletion		Release	Essential
User tag change - add, update and delete		Release	Essential

8.3.1 Presence changed

This trigger receives a notification anytime a user's (aggregate) presence changes.

Table 270: Presence changed trigger - Inputs

Parameters	Description
Who	A drop-down list of all users; select the user you wish to monitor
State	The presence state on which you want to base the trigger. The options are Busy, Away, Available, Any.

Table 271: Presence changed trigger - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	JSON formatted response.

8.3.2 User changed

This trigger will be notified when a user is modified, added or deleted within CloudLink either manually or via a connected app/integration.

Table 272: User changed trigger - Outputs

Parameters	Description
Action	Indicates whether the user was added, modified or updated.
UserId	Unique identifier of the updated user.

8.3.3 User deactivation/deletion

This trigger is activated when a user is deactivated or deleted.

Table 273: User deactivation/deletion trigger - Outputs

Parameters	Description
Action	Indicates whether the user was deactivated or deleted.
UserId	Unique identifier of the updated user.
Result	JSON formatted response.

8.3.4 User tag change - add, update and delete

This trigger is activated when the user tag is added, updated or deleted.




Table 274: User tag change - add, update and delete trigger - Outputs

Parameters	Description
Action	Indicates whether the user was added, updated or deleted.
UserId	Unique identifier of the updated user.
Result	JSON formatted response.

8.4 Integrations Triggers

The following table describes the various triggers supported under **Integrations** category in the **Workflow Canvas**:

Table 275: Integrations Triggers

Trigger Name	Icon	Availability	License Type
Twilio incoming messages		Release	Premier
CM.com incoming messages		Release	Premier
Zoom trigger		Beta	Essential

8.4.1 Twilio incoming messages

This trigger allows you to receive a notification from Twilio when an SMS or WhatsApp message is received. Click the **Copy URL** button to retrieve the webhook URL. This can be pasted into the Twilio console using the number/channel from which you want to receive notifications.

Table 276: Twilio incoming messages trigger - Throughputs

Parameters	Description
Connection	A list of connections defined for the Twilio integration within Workflow Studio. This is the Twilio connection that will be used to send a response to the message from the Workflow, if applicable.
AttachmentConnection	Microsoft Graph connection to optionally be used to store attachments received.
AttachmentFolderName	Name of the folder where incoming attachments will be stored.
CustomerAvatarSVG	File defined in .svg format with the image to be used for the customer avatar in the chat conversation.

Table 277: Twilio incoming messages trigger - Outputs

Parameters	Description
From	Number from which the message is received.
AccountSid	Twilio security identifier for the account.
MessageSid	Twilio security identifier for the message.
To	Number to which the message is sent.
Event	JSON object with information about the message.
NumSegments	Number of message segments.
FromCountry	Country from which the message was sent.
FromCity	City from which the message was sent.
FromState	State or province from which the message was sent.
ToCity	City to which the message was sent.

Parameters	Description
ToState	State or province to which the message was sent.
ToCountry	Country to which the message was sent.
Headers	Array of HTTP headers for the incoming message.
Body	Text body of the message.
SessionId	Unique session identifier.
Message	The received message.

8.4.2 CM.com incoming messages

This trigger receives a notification from CM.com when a message is received. Click the **Copy URL** button to retrieve the webhook URL. This can be pasted into the CM.com console using the number/channel from which you want to receive notifications.

Table 278: CM.com incoming messages trigger - Throughputs

Parameters	Description
Connection	A list of connections defined for the CM.com integration within Workflow Studio. This is the CM.com connection that will be used to send a response to the message from the Workflow, if applicable.
AttachmentConnection	Microsoft Graph connection to optionally be used to store attachments received.
AttachmentFolderName	Name of the folder where incoming attachments will be stored.
CustomerAvatarSVG	File defined in .svg format with the image to be used for the customer avatar in the chat conversation

Table 279: CM.com incoming messages trigger - Outputs

Parameters	Description
From	Number from which the message is received.
To	Number to which the message is sent.
Message	JSON object with information about the message.
Channel	Channel from which the message was sent (SMS, Facebook Messenger or WhatsApp Business).
SessionId	Unique session identifier.

8.4.3 Zoom trigger

This trigger allows users to receive notifications when an event is generated within the Zoom ecosystem. Many events within Zoom can be configured to notify the Zoom trigger that a change has occurred. The URL for the trigger should be copied to the Zoom Application. For detailed instruction on how to configure the trigger notification, refer to the [Configure the Trigger Notification in the Zoom App](#) section.

Table 280: Zoom trigger - Inputs

Parameters	Description
ZoomSecretToken	Token defined in Zoom that will be used to authenticate the message received.





Table 281: Zoom trigger - Outputs

Parameters	Description
Body	JSON object with the message body.
Status	Status information about the event.
Event	String representation of the event in Zoom.

8.5 Developer Triggers

The following table describes the various triggers supported under Developer category in the Workflow Canvas:

Table 282: Developer Triggers

Trigger Name	Icon	Availability	License Type
Webhook		Release	Essential
Sign In App notification		Beta	Essential
Schedule		Release	Essential
SIP DECT notification		Beta	Essential

8.5.1 Webhook

This trigger allows you to receive notifications from an external application; copy the URL within the trigger and paste it into your 3rd-party application.

Webhook triggers support different types of security.

There are two different types of Webhooks:

- **Unauthenticated** - which means that it can be called without providing a token.
- **Authorized** - a valid CL token must be provided in the Authorization header. A webhook is set as Authorized when the Run As is set to a Caller.

By default, this trigger runs asynchronously. Once the trigger event is received, the response is sent immediately, indicating that the workflow has successfully started.

Enable the **Execute synchronously** toggle to run synchronously. If this toggle is selected, the response to the request to start the workflow will not be sent until the workflow completes.

Note:

If the workflow requires more than 20 seconds to complete, it will timeout while running in synchronous mode and report an error.

8.5.2 Sign In App notification

This trigger receives a notification when a notification is received from the SignInApp application (<https://my.signinapp.com/login>). The trigger allows for the verification of a secret key to ensure that the notification is being sent from the registered SignInApp.

Table 283: Sign In App notification trigger - Inputs

Parameters	Description
Secret	Value set in SignInApp used to validate the incoming message.

Table 284: Sign In App notification trigger - Outputs

Parameters	Description
VisitorName	Name of the visitor.
EventType	Indicates the event type. Options are 'signin' and 'signout'.
VisitorCompany	Company associated with the visitor.
VisitorCarReg	Car registration information for the visitor.

8.5.3 Schedule

This trigger receives a notification when the configured time elapses.

Table 285: Schedule trigger - Inputs

Parameters	Description
Timezone	This is a list of time zones.

Parameters	Description
Number	Enter the value corresponding to the selected Duration Type. For example, if Minute is selected for Duration Type, enter 10 to wait 10 minutes.
Duration Type	Dropdown of duration types (minute/ hour/day/week/month/year)
Termination Type	Indicator of when the schedule will terminate. There are two options. <ul style="list-style-type: none"> • Ends On – the flow will terminate on the date entered • Never – the flow will run forever

8.5.4 SIP DECT notification

This trigger receives a notification from an Event Manager with a notification event and text.

Table 286: SIP DECT notification trigger - Throughputs





Parameters	Description
EventBody	JSON object with information about the event generated.
EventManager	JSON object with information about the Event Manager that generated the event.
Event	JSON object with details about the event.
NotificationText	Text information describing the event.
CallbackNumber	Number to be called back; applicable for some event types.
SourceEndpoint	Information about the endpoint generating the event.
TargetEndpoints	List of endpoints that are receiving the event notification.
EventPlan	Object with additional details about the event.

Parameters	Description
ReqType	Request type for the event.
apiKey	Unique key assigned to the Event Manager.

8.6 Search Only Triggers

The following table describes the various uncategorized triggers supported in the **Workflow Canvas** and only available through the **Search**/more option in the center of the activity wheel:

Table 287: Search Only Triggers

Trigger Name	Icon	Availability	License Type
Slack webhook		Beta	Essential
Presence source changed		Beta	Essential
Github issues		Beta	Essential
VCCS Trigger		Beta	Essential

8.6.1 Slack webhook

This trigger receives an incoming message from a Slack channel.

Table 288: Slack webhook trigger - Inputs

Parameters	Description
Token	This value can be found when creating the outgoing webhook trigger via Slack.
Channel	The slack channel(s) that you would like to listen to; '*' denotes all channels.
Username	Username of the user sending the message; '*' denotes all users.

Table 289: Slack webhook trigger - Outputs

Parameters	Description
Result	JSON formatted result object.
Text	Message received.

8.6.2 Presence source changed

This trigger receives a notification for the selected user when the monitored user's presence changes for any presence category (there are many different categories, which include phones, ACD, calendar, and apps). If the user sets their presence to busy and then receives a call, this trigger will be notified as the underlying phone extension changed from available to busy; however, their aggregate presence will still display busy as it was before the call.

Table 290: Presence source changed trigger - Throughputs

Parameters	Description
Who	Specifies the unique identifier for the user.
Status	Options are 'Any', 'Away', 'Busy' or 'Available'. Indicates which updates will enable the trigger.
Source	Indicates the source that caused the presence change

Table 291: Presence source changed trigger - Outputs

Parameters	Description
Success	Indicates whether the activity was completed successfully (value = True) or not (value = False).
Result	Returns the JSON formatted response from the activity.

8.6.3 Github issues

This trigger receives a notification when an event from Github is sent. Since this trigger is a custom webhook, copy the URL to the third-party application. Refer to the [GitHub Integration](#) section for instructions on how to set up Github Integration.

Table 292: Github issues trigger - Inputs

Parameters	Description
Connection	Github connection

Table 293: Github issues trigger - Outputs

Parameters	Description
Action	Action taken on Github.
EventBody	JSON formatted object with information about the event.
Repository	Name of the Github repository.
Labels	Array of labels associated with the Github issue.
Title	Title of the Github issue.
Number	Github issue number.
Url	URL of the Github issue.
State	Current state of the Github issue.

Parameters	Description
IsComment	Indicates if the update is due to a comment update.
Comment	Comment for the Github issue.

8.6.4 VCCS Trigger

This trigger receives a notification from VCCS.

Table 294: VCCS Trigger - Outputs

Parameters	Description
AppointmentId	Unique identifier for the appointment
Action	Action taken for the appointment. Refer to the VCCS documentation for details on action types.
Message	Message sent with the appointment
ScheduledTime	Scheduled time in GMT for the appointment
Result	JSON response with additional information about the message.
Address	Address associated with the appointment

This chapter contains the following sections:

- [Anthropic Integration \(Premier\)](#)
- [CM.com Integration \(Premier\)](#)
- [Edge Signal Integration \(Premier\)](#)
- [GitHub Integration \(Essential\)](#)
- [Google Integration \(Essential\)](#)
- [Google Gemini Integration \(Premier\)](#)
- [HTTP Integration \(Premier\)](#)
- [Jira Cloud Integration \(Premier\)](#)
- [Microsoft Azure Integration \(Premier\)](#)
- [Microsoft Graph Integration \(Premier\)](#)
- [OpenAI Assistants Integration \(Premier\)](#)
- [Outlook Integration \(Essential\)](#)
- [Salesforce Integration \(Premier\)](#)
- [Twilio Integration \(Premier\)](#)
- [Whisper Integration \(Premier\)](#)
- [Zoom Integration \(Essential\)](#)

Mitel Workflow Studio supports integration with other applications, such as MiCollab, MiContact Center, CM.com, Outlook 365, OpenAI, etc.

9.1 Anthropic Integration (Premier)

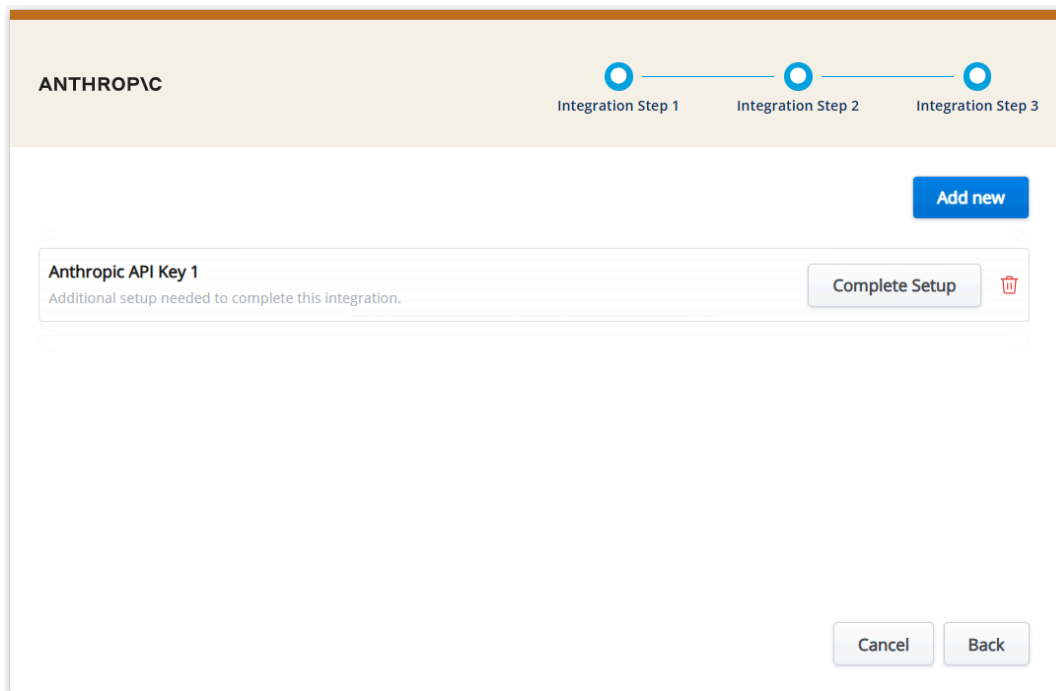
Anthropic Integration requires an API Key that is generated in the Anthropic application. For instructions on how to retrieve the API Key, refer to the [Anthropic](#) documentation.

9.1.1 Create Anthropic Integration

To create a Anthropic Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click on the **See details** button in the **Anthropic** integration tile.
4. Click **Connect** to start the integration steps.
5. In the **Integration Step 1** screen, read through the Getting Started section and click **Next**.

6. In the **Integration Step 2** screen, click **Add New** to create a new connection.



ANTHROPIC

Integration Step 1 Integration Step 2 Integration Step 3

Add new

Anthropic API Key 1

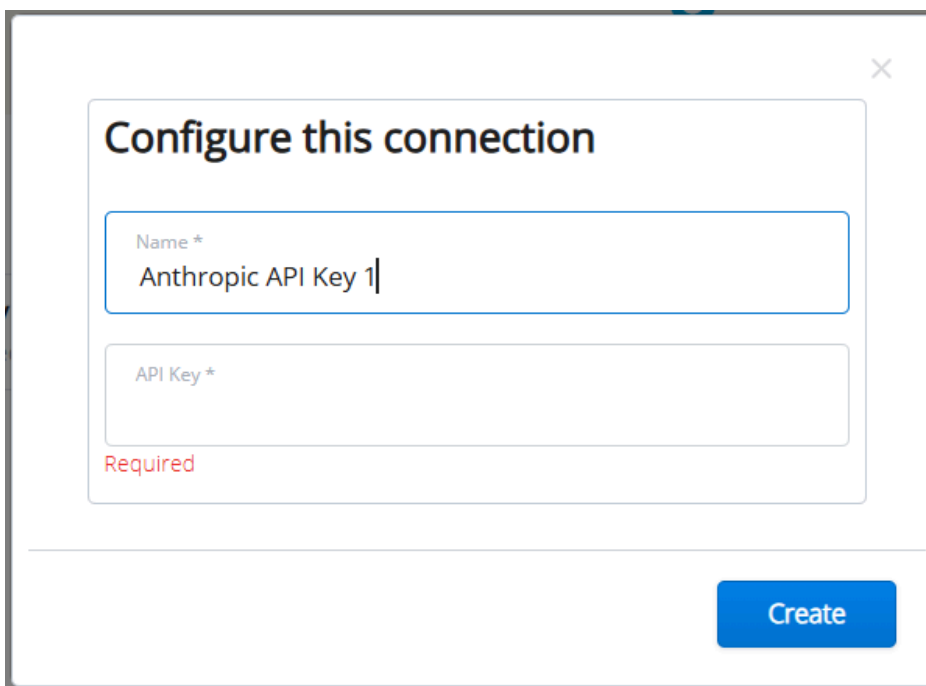
Additional setup needed to complete this integration.

Complete Setup

Cancel Back

Figure 69: Anthropic Integration - Add new

7. Click **Complete Setup** to provide a connection name and the API key provided by Anthropic.



×

Configure this connection

Name *

Anthropic API Key 1

API Key *

Required

Create

Figure 70: Anthropic Integration - Configure this connection

8. Click **Create**.




The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

9. Click **Continue**.

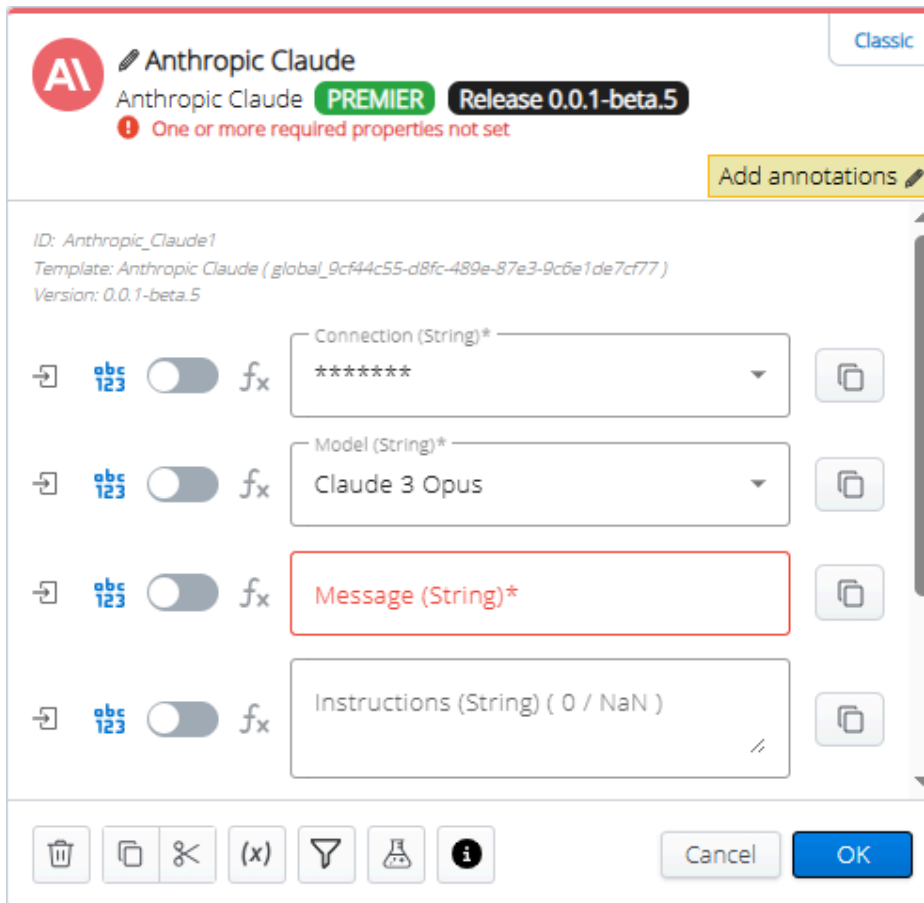
The newly created connection is displayed in the list.

9.1.2 Add an Anthropic activity

To add Anthropic Claude activity in a flow, perform the following steps.

1. Navigate to the **Workflow Editor** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click the  icon to add the new activity to the flow.
6. Under **Integration** activities, select the , click  to add the Anthropic Claude activity to the flow.

The **Anthropic Claude** pop up screen is displayed.



Anthropic Claude Classic

Anthropic Claude **PREMIER** **Release 0.0.1-beta.5**

One or more required properties not set

Add annotations

ID: Anthropic_Claude1
Template: Anthropic Claude (global_9cf44c55-d8fc-489e-87e3-9c6e1de7cf77)
Version: 0.0.1-beta.5

Connection (String)*

Model (String)*
Claude 3 Opus

Message (String)*

Instructions (String) (0 / NaN)

Cancel OK

Figure 71: Anthropic Claude

7. In the **Anthropic Claude** activity screen, provide the configuration options as described in the Anthropic Claude field description table.

Note:
Mandatory fields are marked with an asterisk (*).

Table 295: Anthropic Claude – Field Description

Field	Description
Connection*	Select the connection created in the Anthropic integration from the drop-down list.
Model*	Enter the AI model to use.
Message*	Enter the user message for sending to the AI model.
Instructions	Enter any specific instructions you want to provide to the AI model.

8. Click **OK** to add an Anthropic Claude activity to the flow.

9.1.3 Anthropic Claude Use Case

To use the Anthropic Claude activity within **Workflow Studio**, you can use voice or chat media activities. The example below shows how to route the caller to the correct queue based on their reason for calling.

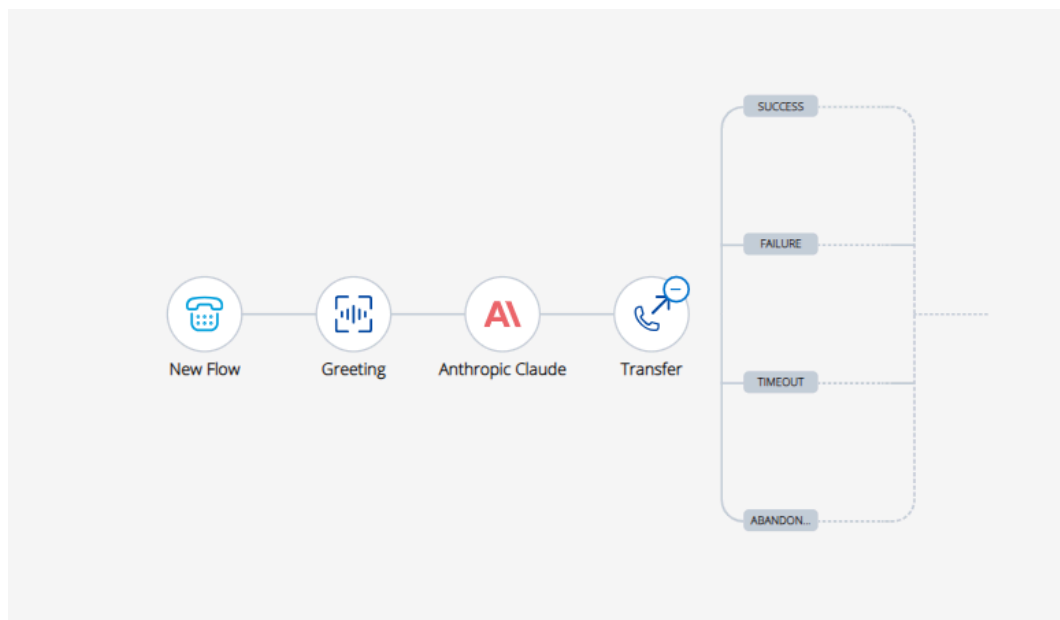


Figure 72: Anthropic Claude Use Case

To create the sample workflow for the above-mentioned use case, perform the following steps:


1. In the **Workflow Editor**, add a **Ringing Endpoint** trigger.
2. Add an **Anthropic Claude** activity to ask the configured integration AI model which queue to send them to.
3. Add a **Transfer** activity to transfer the caller to the queue returned from the AI model's response.

9.2 CM.com Integration (Premier)

CM.com Integration requires the **Product Token key** located in your CM.com account. Refer to the [CM.com](#) documentation for instructions on how to retrieve the token.

9.2.1 Create CM.com Integration

To create CM.com Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integration Hubs**, click  icon on the **CM.com** integration tile.
4. Click **Add new** to create a new connection.

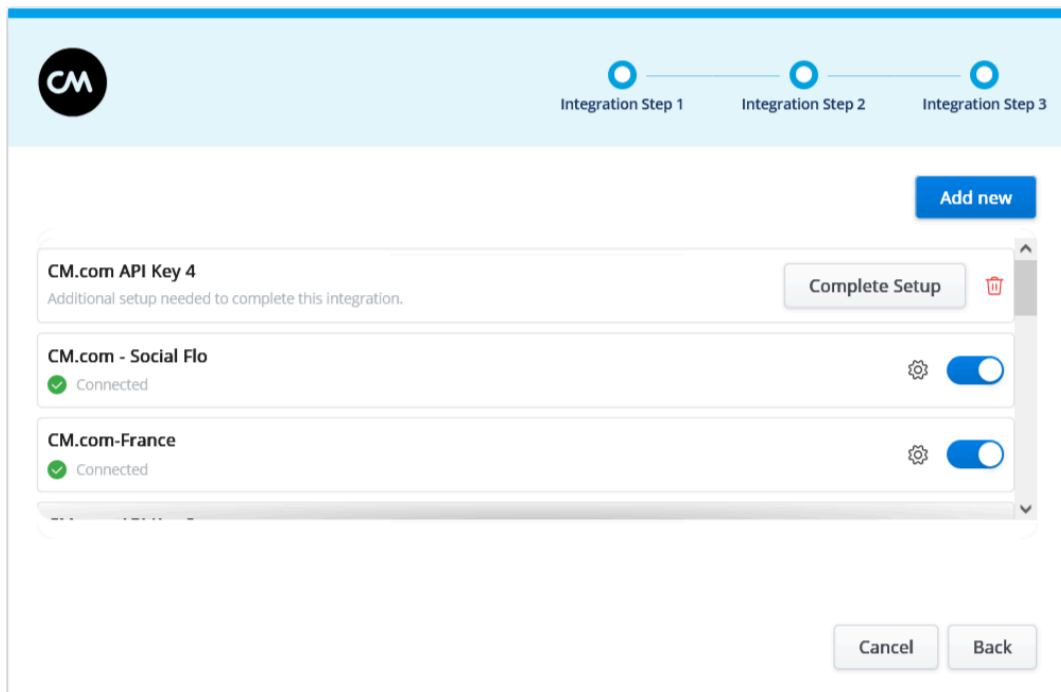


Figure 73: CM.com Integration - Add new

5. Click **Complete Setup** to provide a connection name and the product token obtained from the CM.com website.

Figure 74: CM.com Integration - Configure this connection

6. Click **Create**.


The **Connection Complete** screen is displayed to confirm the success of the third-party integration.



7. Click **Continue**.

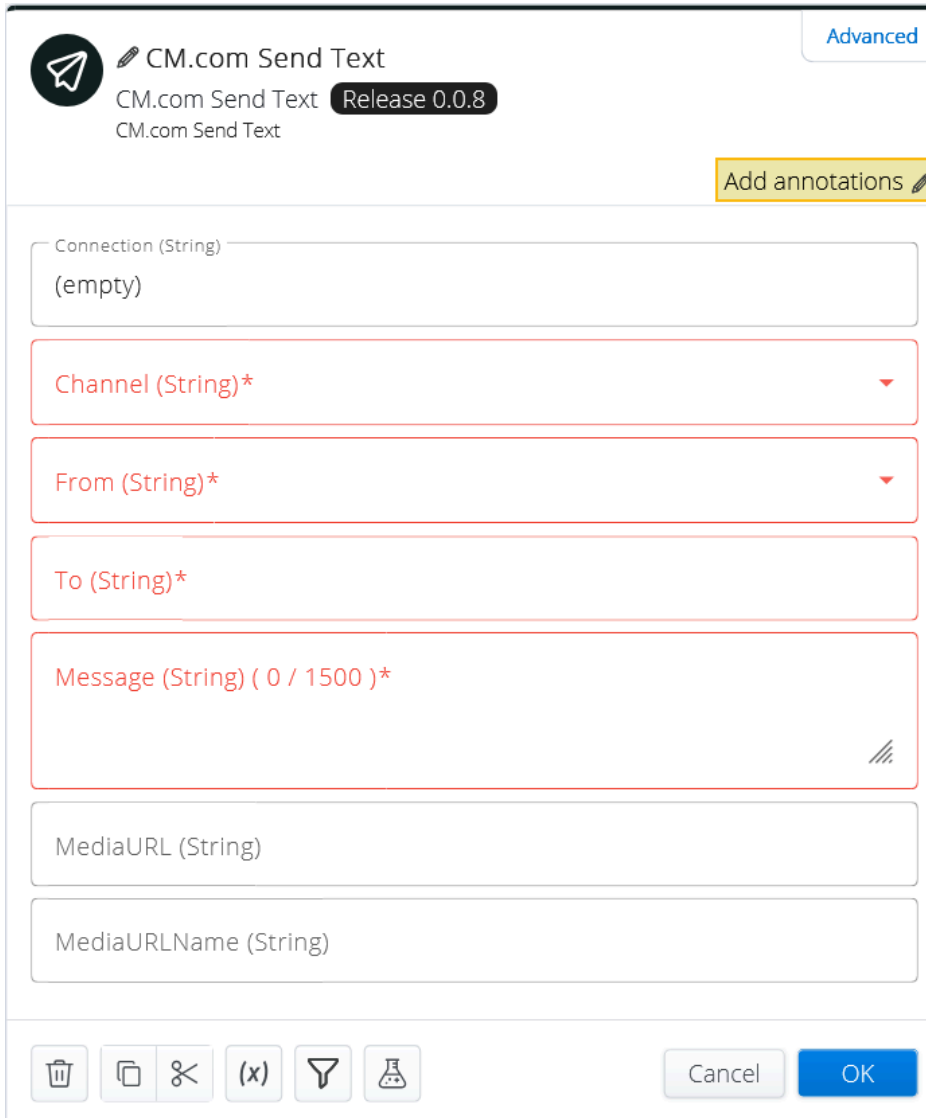
The newly created connection is displayed in the list.

9.2.2 Add a CM.com Send Message activity

To add a Send Message activity to a flow, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click  icon to add the new activity to the flow.

6. Under **Integration** activities, under  icon, click  to add Send Message activity to the flow. The **Send Message Activity** popup screen is displayed.



CM.com Send Text
CM.com Send Text Release 0.0.8

Advanced

Add annotations

Connection (String)
(empty)

Channel (String)*

From (String)*

To (String)*

Message (String) (0 / 1500)*

MediaURL (String)

MediaURLName (String)

Cancel OK

Figure 75: Send Message Activity

7. On the **Send Message Activity** screen, provide the following details. For more information on fields, refer to the field description table.

**Note:**

Mandatory fields are marked with an asterisk (*).

Table 296: Send Message Activity – Field Description

Field	Description
Connection	Select the connection created in the CM.com integration from the drop-down list.
Channel	Select the media type used (WhatsApp / SMS / Facebook)
From	Select the registered mobile number from the drop-down list. The list is pre-populated based on the selected channel.
To	Enter the destination phone number (E.164 format). The value will vary depending on the selected channel. <ul style="list-style-type: none"> • SMS - E.164 number (for example: +16135551234) • WhatsApp - E.164 number <p>Note: Based on WhatsApp's security policy, you can only send a message to a WhatsApp number after receiving one within 24 hours.</p> <ul style="list-style-type: none"> • Facebook - Session Id
Message	Enter the message string/function that you would like to send to the recipient.
MediaURL	Enter the URL of the media that you want to send. It must be publicly accessible.
MediaURLName	Enter the display name of the media when the recipient receives it.

8. Click **OK** to add the CM.com Send Message activity to the flow.

9.3 Edge Signal Integration (Premier)

Edge Signal Integration is required to use the Edge Signal activity and trigger in Workflow Studio.

Prerequisites

To integrate Workflow Studio with Edge Signal, you must create an API Key.

Log in to your account at <https://platform.edgesignal.ai> to configure your account and generate the API key.

9.3.1 Create Edge Signal Integration

To integrate Edge Signal into **Workflow Studio**, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.

The **Integrations Hub** screen is displayed.

3. In the **Integrations Hub**, click **See details** next to the **Edge Signal** integration tile.
4. Click **Connect**.
5. Click **Next**.

The **Edge Signal Integration - Add new connection** screen is displayed.

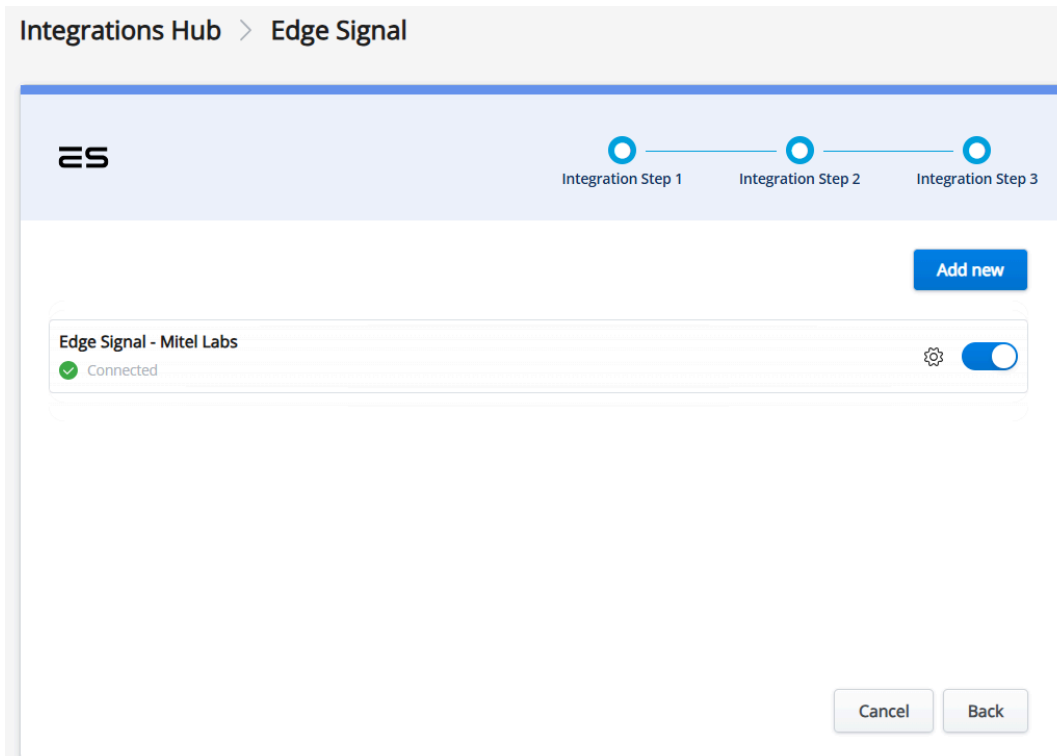


Figure 76: Edge Signal Integration – Add new

6. Click **Add new** to create a new connection. A new API Key connection for Edge Signal will be added.

7. Click **Complete Setup**.

The **Edge Signal Integration - Configure this connection** screen is displayed.

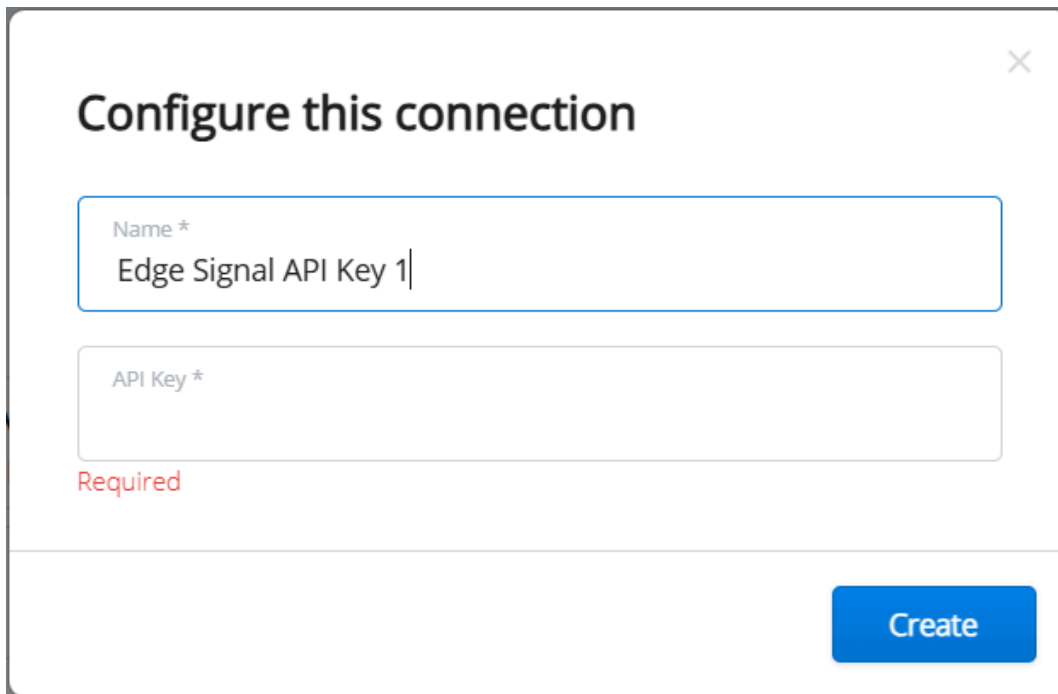


Figure 77: Edge Signal Integration - Configure this connection

8. In the **Name** field, enter the descriptive name that helps you to identify the connection while adding the activity.
9. In the **API Key** field, enter the API Key value obtained from Edge Signal.
10. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

11. Click **Continue**.

The connection is now displayed in the list.

If the integration does not display as Connected, click on the gear icon and confirm that the API Key is entered correctly.

9.3.2 Add an Edge Signal Trigger

Two triggers are available for Edge Signal:

- **Edge Signal trigger:** Allows you to trigger using an Edge Signal device. Details about the device and its action are returned in the EdgeSignalObject output variable.
- **Edge Signal people counter trigger:** Allows you to trigger on the activity of people moving in or out of an area. Details about the activity type and movement information are returned in the trigger output variables.

Refer to the **Edge Signal API documentation** for more details about the information returned in these triggers.

To invoke the triggers, click the **Copy** button next to **Copy URL to invoke Webhook**, as shown below, to copy the URL from the Edge Signal trigger.

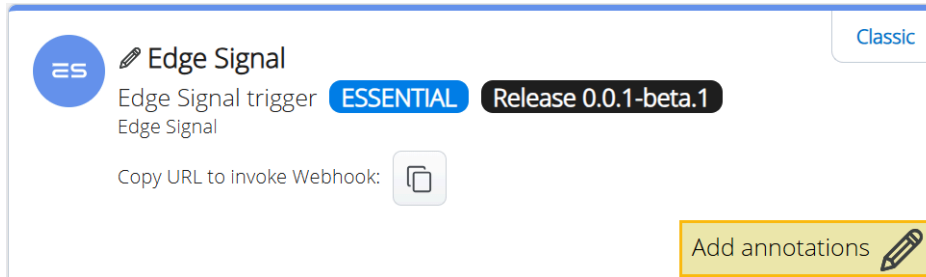


Figure 78: Edge Signal Trigger

Add the URL to Edge Signal (<https://platform.edgesignal.ai>) to invoke the Webhook when activity occurs, and the trigger will be invoked.

9.3.3 Add an Edge Signal Activity

After Edge Signal integration is added, the **Edge Signal shell command** activity can be added to a workflow. This activity allows the workflow to invoke a command for a selected Edge Signal system.

1. Navigate to the **Workflow Canvas** using **Create flow**.
2. Search and select the Edge Signal trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click the **Add** icon to add the new activity to the flow.

6. Click the **Search** icon to search and add the **Edge Signal shell command** activity to the flow.

The **Edge Signal shell command activity** popup screen is displayed.

The screenshot shows the configuration window for the 'Edge Signal shell command' activity. At the top, it identifies the activity as 'ESSENTIAL' and in 'Release 0.0.1-beta.2'. A red warning icon and text indicate that 'One or more required properties not set'. The main area contains several input fields, each preceded by a small icon (a square with 'ab' and '123') and a function icon 'fx'. The fields are: 'Connection (String)*' with a dropdown menu showing 'Edge Signal - Mitel ...', 'AccountId (String)*', 'ProjectId (String)*', and 'ShellCommandTopic (String)*'. Each field has a copy icon to its right. At the bottom of the window, there is a toolbar with icons for deleting, copying, pasting, undo, redo, and a help/info icon, followed by 'Cancel' and 'OK' buttons.

Figure 79: Edge Signal shell command activity

7. In the **Edge Signal shell command activity** screen, provide the following details.
 - a. **AccountId**
 - b. **ProjectId**
 - c. **ShellCommandTopic**
 - d. **ShellCommandMessage**.
 8. Enter the list of one or more **TargetDeviceIds** in the input fields. For details on these values, refer to the **Edge Signal API** description.
- The **Success** output variable will indicate whether the command was successfully received and executed by the Edge Signal activity or trigger.
9. Click **OK** to add the **Edge Signal shell command** activity to the flow.

9.4 GitHub Integration (Essential)

GitHub Integration is required to use the GitHub activities in **Workflow Studio**.

9.4.1 GitHub Requirements

To integrate Workflow Studio with GitHub, you must have a GitHub account.

1. Log into your GitHub account to create a personal API Key that will be used to perform actions within Workflow Studio.
2. To retrieve/create an API key, select your Profile.
3. Click **Settings** and then, click **Developer Settings**.
4. Under **Developer Settings**, click **Personal Access Tokens**.
5. Click **Generate new token**.

Once the token is created, store it in a safe and secure location.

9.4.2 Create GitHub Integration

To create GitHub integration into **Workflow Studio**, perform the following steps:

1. Log in to **Workflow Studio**.
2. Under **Library**, click **Integrations**.

The **Integrations Hub** screen is displayed.

3. In the **Integrations Hub** screen, click **See details** next to the **GitHub** integration tile.
4. Click **Connect**.
5. Click **Next**.

The **GitHub Integration - Add new** screen is displayed.

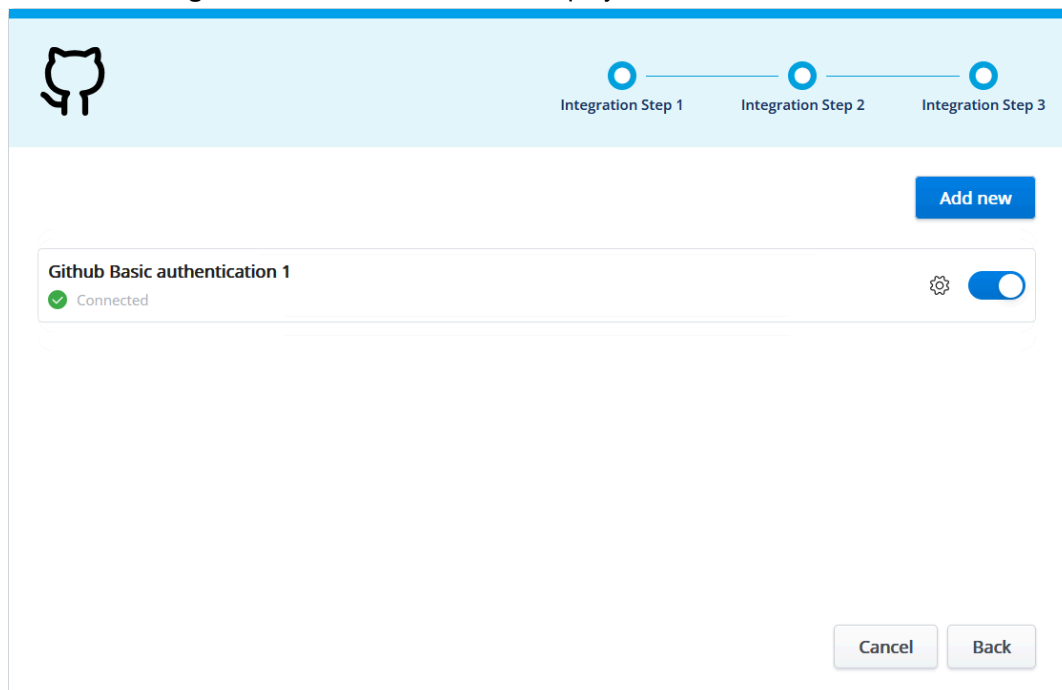
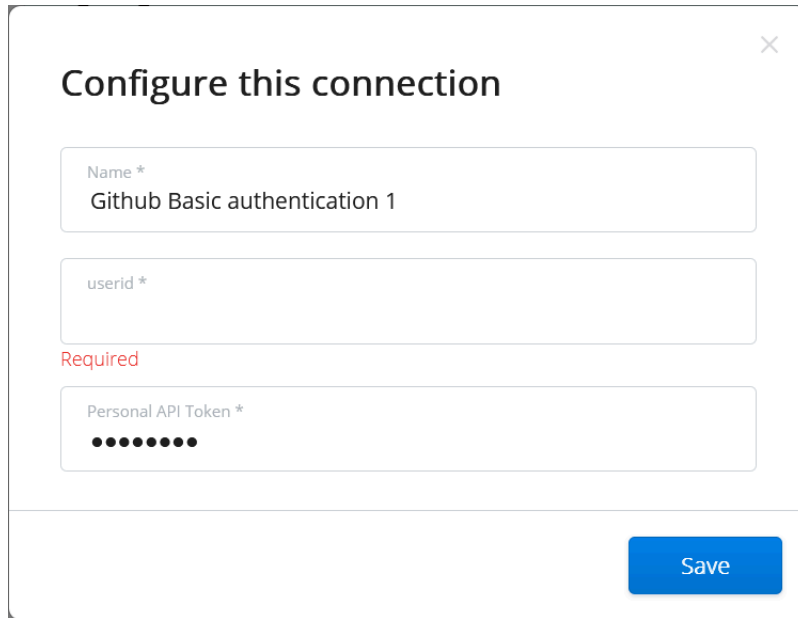


Figure 80: GitHub Integration - Add new

6. Click **Add new** to add a new connection.
7. Click **Complete Setup** to create a new connection.

The **GitHub Integration - Configure this connection** screen is displayed.



The screenshot shows a modal window titled "Configure this connection" with a close button (X) in the top right corner. Inside the modal, there are three input fields: "Name *" with the text "Github Basic authentication 1", "userid *" which is empty, and "Personal API Token *" which contains a series of dots. Below the "userid *" field, the word "Required" is written in red. At the bottom right of the modal is a blue "Save" button.

Figure 81: GitHub Integration - Configure this connection

8. In the **Configure this connection** screen, enter the following details.
 - a. In the **Name** field, enter a descriptive name for the connection.
 - b. In the **Userid** field, enter your GitHub username.
 - c. In the **Personal API Token** field, enter the token created and stored in the requirement steps.
9. Click **Save**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

9.4.3 Use the GitHub API in Workflow

To make REST API calls into GitHub from a workflow, perform the following steps:

1. In the **Workflow Studio**, open the new or existing flow to which you want to add the REST API call activity.

2. In the **Workflow Canvas**, open the activity wheel.



Figure 82: Search Activity Wheel

3. Under the **Developer** category, click on the **HTTP Request** activity.

The **HTTP Request** activity screen is displayed.

Figure 83: HTTP Request

4. In the **Http request** activity, click on the **Authorization** tab.
5. In the **Integration** field, select the **Github** option from the list.
6. In the **Connection** field, select the connection you created from the list.
7. Enter the GitHub API you will be using.

In the above example, **GET https://api.github.com/issues** is used to retrieve all issues in the account.

8. Click on the **Testing** tab to confirm that the connection is set up correctly to call the API.

The results appear in the **Response** area of the tab.

If there is any error, check the details of the error returned and verify that the application added in GitHub has the appropriate scope value to access the called API.

9.5 Google Integration (Essential)

Google Integration is required to use the Google activities and API in Workflow Studio.

9.5.1 Prerequisites

To integrate Workflow Studio with Google, you must set up a unique Google security identifier, which is either an OAuth 2.0 client ID or an API key.

To create a unique identifier:

1. Log into the Google API Console (<https://console.developers.google.com>).
2. Select an existing project or create a new one.
3. Select **APIs & services** from the left-hand side console menu.
4. Select **Credentials**.
5. Select **Create Credentials**.
6. Select either **API key** or **OAuth client ID**.
7. Complete the steps in the Google UI to create an API key or OAuth client identifier.

9.5.2 Create Google Integration

To integrate Google into **Workflow Studio**, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.

The **Integrations Hub** screen is displayed.

3. In the **Integrations Hub**, click **See details** next to the **Google** integration tile.
4. Click **Connect**.

5. Click **Next**.

The **Google Integration - Add new connection** screen is displayed.



Figure 84: Google Integration – Add new

6. Click **Add new** to add a new connection using **OAuth2 Authorization Code**, **OAuth2 JWT Bearer**, or **API Key**.

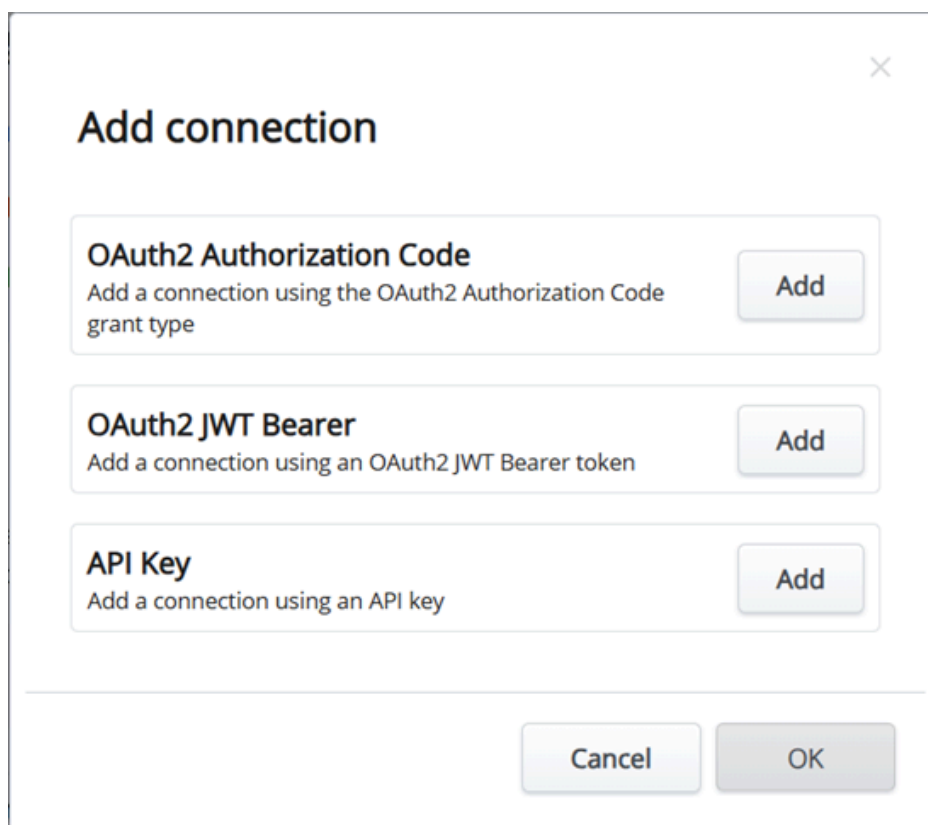


Figure 85: Google Integration – Add connection

The **OAuth2** options are used to access the server resources, such as **Google Drive – File upload** or **Google Drive – Get file contents** actions. If Google APIs that require access to user data are called, an OAuth2 connection is required.

- The **OAuth2 Authorization Code** option allows the application to access resources on behalf of a user.

- The **OAuth2 JWT Bearer** option allows the application to access resources without user interaction. This option is used when accessing resources that are not user-specific.
- The **API Key** option can be used to enable access to **Google Places** within the **Schedule** activity, or any other Google APIs that access public data, such as **Google Maps** or **Google Translate**.

To add the connection using the **OAuth2 Authorization Code** option, perform the following steps:

1. Click **Add** next to **OAuth2 Authorization Code** and click **OK**.
2. Click **Complete Setup**.

The **OAuth2 Authorization Code - Configure this connection** screen is displayed.

Figure 86: OAuth2 Authorization Code – Configure this connection

3. In the **OAuth2 Authorization Code - Configure this connection** screen, enter the `client_id` and `client_secret` available in the Google API Console when setting up OAuth 2.0.
4. Enter the scope value.

The scope value will vary depending on what is required to be accessed by the connection. For more details on scope values, refer to <https://developers.google.com/identity/protocols/oauth2/scopes>.

5. Click **Create**.

The **Google Login** page is displayed to authenticate the entered information for the connection.

6. Complete the authentication in Google page.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

7. Click **Continue**.

The connection is now displayed in the list.

To add the connection using the **OAuth2 JWT Bearer** option, perform the following steps:

1. Click **Add** next to **OAuth2 JWT Bearer** and click **OK**.
2. Click **Complete Setup**.

The **OAuth2 JWT Bearer - Configure this connection** screen is displayed.

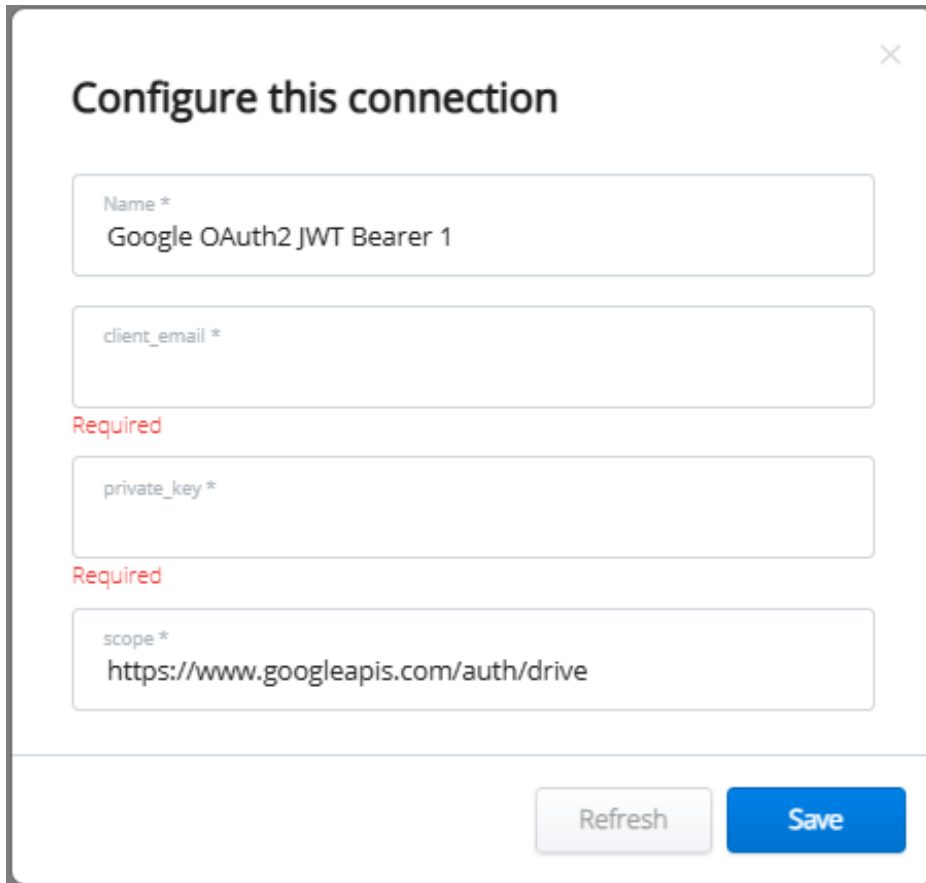
The screenshot shows a modal window titled "Configure this connection" with a close button (X) in the top right corner. Inside the modal, there are four text input fields. The first field is labeled "Name *" and contains the text "Google OAuth2 JWT Bearer 1". The second field is labeled "client_email *" and is empty. Below this field is a red "Required" label. The third field is labeled "private_key *" and is empty. Below this field is a red "Required" label. The fourth field is labeled "scope *" and contains the text "https://www.googleapis.com/auth/drive". At the bottom right of the modal, there are two buttons: a "Refresh" button and a blue "Save" button.

Figure 87: OAuth2 JWT Bearer – Configure this connection

3. Enter the `client_email` and `private_key` values provided from the Google API Console. Note that these values should refer to the service account email, not a specific user email.
4. Enter the scope value.

The scope value will vary depending on what is required to be accessed by the connection. For more details on scope values, refer to <https://developers.google.com/identity/protocols/oauth2/scopes>.

5. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

6. Click **Continue**.

The connection is now displayed in the list.

To add the connection using the **API Key** option, perform the following steps:

1. Click **Add** next to **API Key** and click **OK**.
2. Click **Complete Setup**.

The **API Key - Configure this connection** screen is displayed.

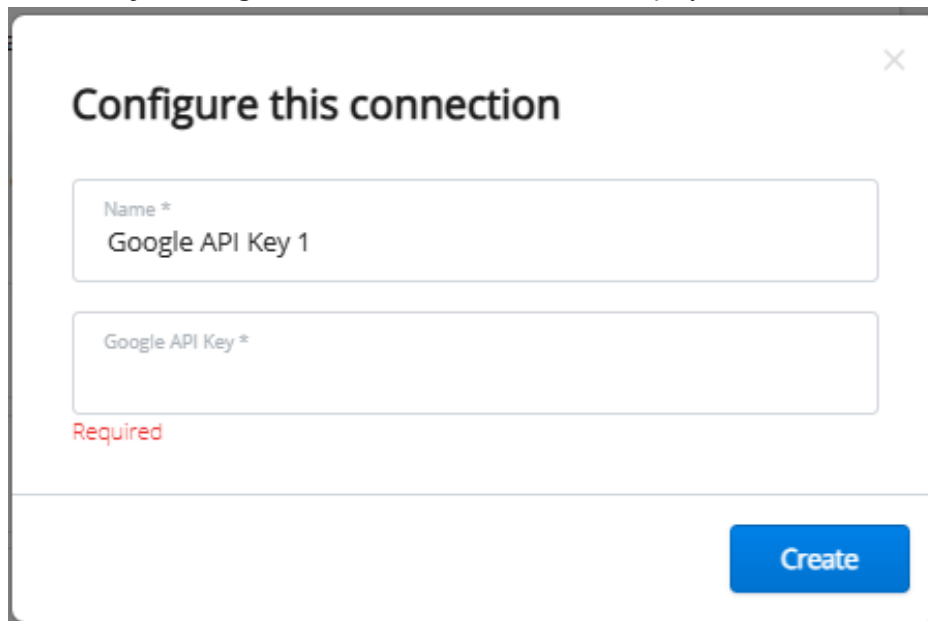
The image shows a dialog box titled "Configure this connection" with a close button (X) in the top right corner. Inside the dialog, there are two text input fields. The first field is labeled "Name *" and contains the text "Google API Key 1". The second field is labeled "Google API Key *" and is currently empty. Below the second field, the word "Required" is written in red. At the bottom right of the dialog, there is a blue button labeled "Create".

Figure 88: API Key – Configure this connection

3. Enter the Google API Key value provided from the Google API Console when setting up the API key.
4. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

5. Click **Continue**.

The connection is now displayed in the list.

If the integration does not display as **Connected**, click on the gear icon and confirm that all the settings are entered correctly.

9.5.3 Use Google APIs in Workflow

To make REST API calls into Google from a workflow, perform the following steps:

1. In **Workflow Studio**, open the new or existing flow to which you want to add the REST API call activity.
2. Open the activity wheel.

3. Under the **Developer** category, click on the **HTTP Request** activity.

The **HTTP Request** activity screen is displayed.

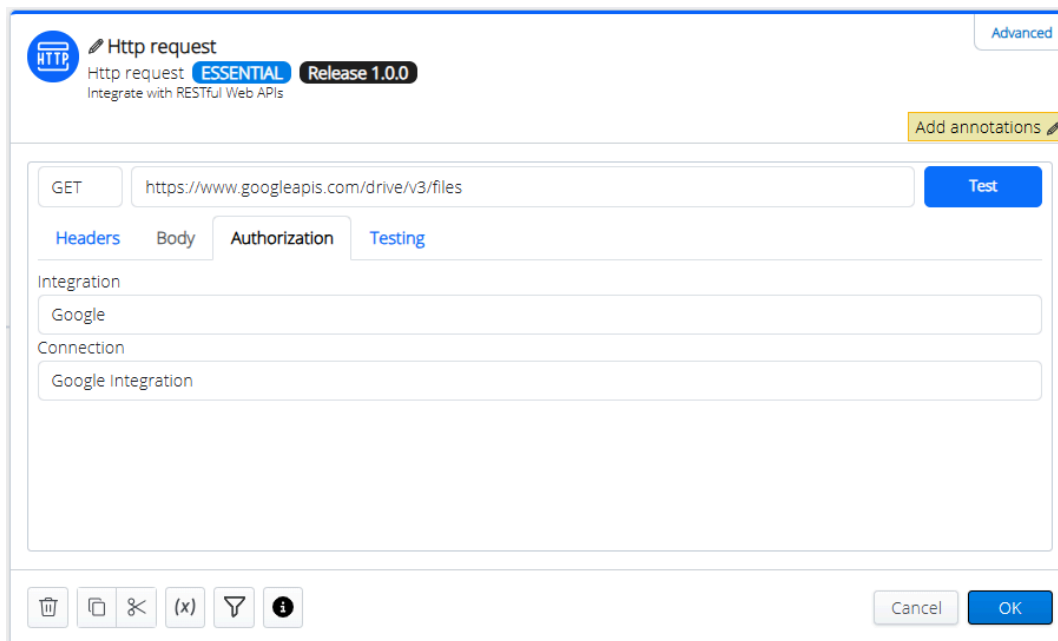


Figure 89: HTTP Request

4. In the **Http request** activity, click on the **Authorization** tab.
5. In the **Integration** field, select **Google** from the option list.
6. In the **Connection** field, select the connection you created from the list.
7. Enter the Google API you will be using.

In the above example, **GET https://www.googleapis.com/drive/v3/files** is used to retrieve the file list from Google Drive. For a complete list of Google APIs, refer to <https://developers.google.com/apis-explorer>.

8. Click on the **Testing** tab to confirm that the connection is set up correctly to call the API.

The results appear in the **Response** area of the tab.

If an error occurs, check the details of the error returned and verify that the Google connection configured in Workflow has the appropriate scope value to access the API called.

9.5.4 Use Google with Workflow Activities

The following activities require a Google integration connection to be configured:

Schedule


To automatically retrieve the business hours of an organization within the **Schedule** activity, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create flow**.
2. Open the activity wheel.

- Under the **Routing** category, click on the **Schedule** activity.

The **Schedule** activity screen is displayed.

Figure 90: Schedule

- Enable the **Use Google Places** checkbox.
- From the **Connection** drop-down list, select the configured integration connection for Google. Note that this can be an APIKey or OAuth2 connection.
- Enter the **Place ID** for the business or select the  icon to retrieve the Place ID using the name and address of the business. If the business information is returned correctly, the name and address will display, as well as the scheduled open/closed hours for the business.
- Click **OK**.

Google Drive - File upload

This activity can be used to upload a file from a shared URL to a specified location on a Google Drive. The API <https://www.googleapis.com/upload/drive/v3/files> is used to upload the file.

- Navigate to the **Workflow Canvas** using **Create flow**.
- Open the activity wheel.

- Under the **Integrations** category from the activity wheel, select **Google** and select **Google Drive - File upload**.

Google Drive - File upload Classic

Google Drive - File upload **ESSENTIAL** **Release 0.0.1-beta.3**

This activity will upload a file from a shared URL to a location specified on a Google Drive.

❗ One or more required properties not set

Add annotations

ID: Google_Drive__File_upload1
Template: Google Drive - File upload (global_68fe75ed-030a-4771-903c-4bd0fc7a2e1c)
Version: 0.0.1-beta.3

FolderName (String)

FileURLLocation (String)*

Connection (Object)*

FileName (String)*

Cancel OK

Figure 91: Google Drive – File upload

- In the **FolderName** field, enter an optional folder name.

If specified, the file will be uploaded to that folder. If the folder doesn't exist, the folder will be created in the Google Drive, and the file will be uploaded to that folder. Otherwise, the file will be uploaded to the root directory on the specified Google Drive.

- In the **FileURLLocation** field, enter the shared URL where the file is located.
- From the **Connection** list, select the configured integration connection for Google. Note that this should be an OAuth2 connection.
- In the **FileName** field, enter the name of the file.
- Click **OK**.

Google Drive – Get file contents

This activity can be used to retrieve the contents of a .pdf, .docx, or .txt file stored in a Google Drive. The API <https://www.googleapis.com/drive/v3/files> is used to retrieve the file.

- Navigate to the **Workflow Canvas** using **Create flow**.
- Open the activity wheel.

- Under the **Integrations** category from the activity wheel, select **Google** and select **Google Drive - Get file contents**.

Figure 92: Google Drive – Get file contents

- From the **Connection** list, select the configured integration connection for Google. Note that this should be an OAuth2 connection.
- In the **FileList** field, enter the name(s) of the file(s) to be retrieved.

The **Results** output variable will contain the concatenation of the contents of all the requested files.

- Click **OK**.

9.6 Google Gemini Integration (Premier)

Google Gemini Integration requires an API Key that is generated in the Google AI Studio application. For instructions on how to retrieve the API Key, refer to the [Google Gemini](#) documentation.

9.6.1 Create Google Gemini Integration

To create Google Gemini Integration in Workflow Studio, perform the following steps:

- Log into **Workflow Studio**.
- Under **Library**, click **Integrations**.

3. In the **Integrations Hub**, click on the **See details** button in the **Google Gemini** integration tile.
4. Click **Connect** to start the integration steps.
5. In the **Integration Step 1** screen, read through the Getting Started section and click **Next**.
6. In the **Integration Step 2** screen, click **Add New** to create a new connection.

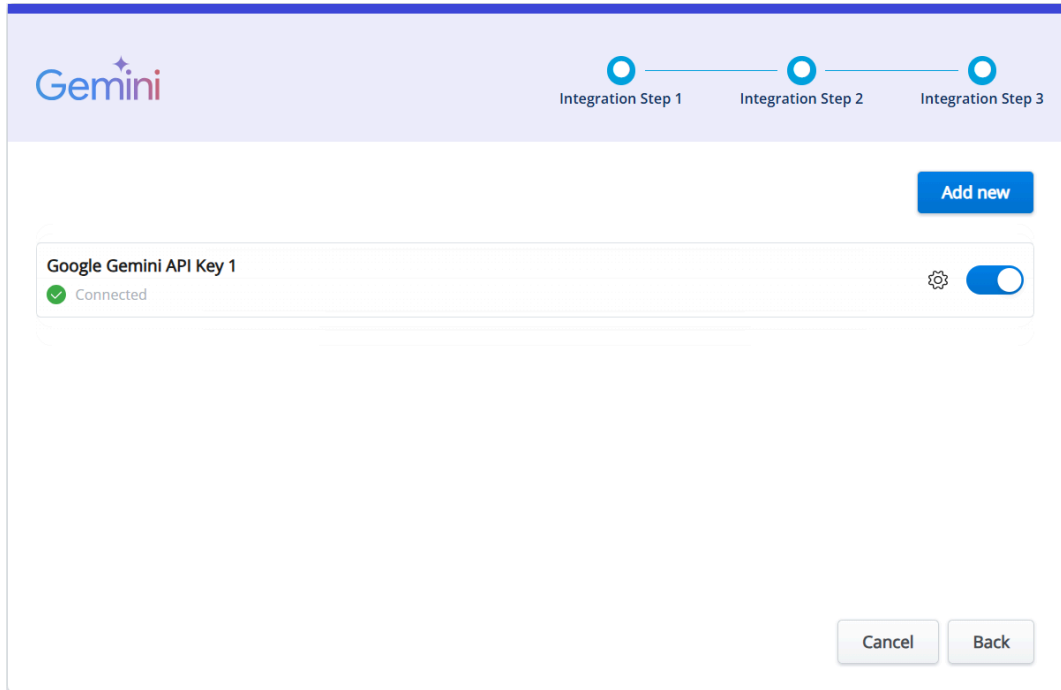


Figure 93: Google Gemini Integration - Add new

7. Click **Complete Setup** to provide a connection name and the API key provided by Google Gemini.

Figure 94: Google Gemini - Configure this connection

8. Click **Continue**.




The newly created connection is displayed in the list.

9. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

9.6.2 Add a Google Gemini Activity

To add Google Gemini activity in a flow, perform the following steps.

- 1. Navigate to the **Workflow Editor** using **Create flow**.
- 2. Search and select a trigger in the **Search activities** search bar.
- 3. Click **Add**.
- 4. Update the required parameters and click **OK**.
- 5. Click  icon to add the new activity to the flow.
- 6. Under **Integration** activities, select the  icon, click  to add Google Gemini activity to the flow.

The **Google Gemini** pop up screen is displayed.

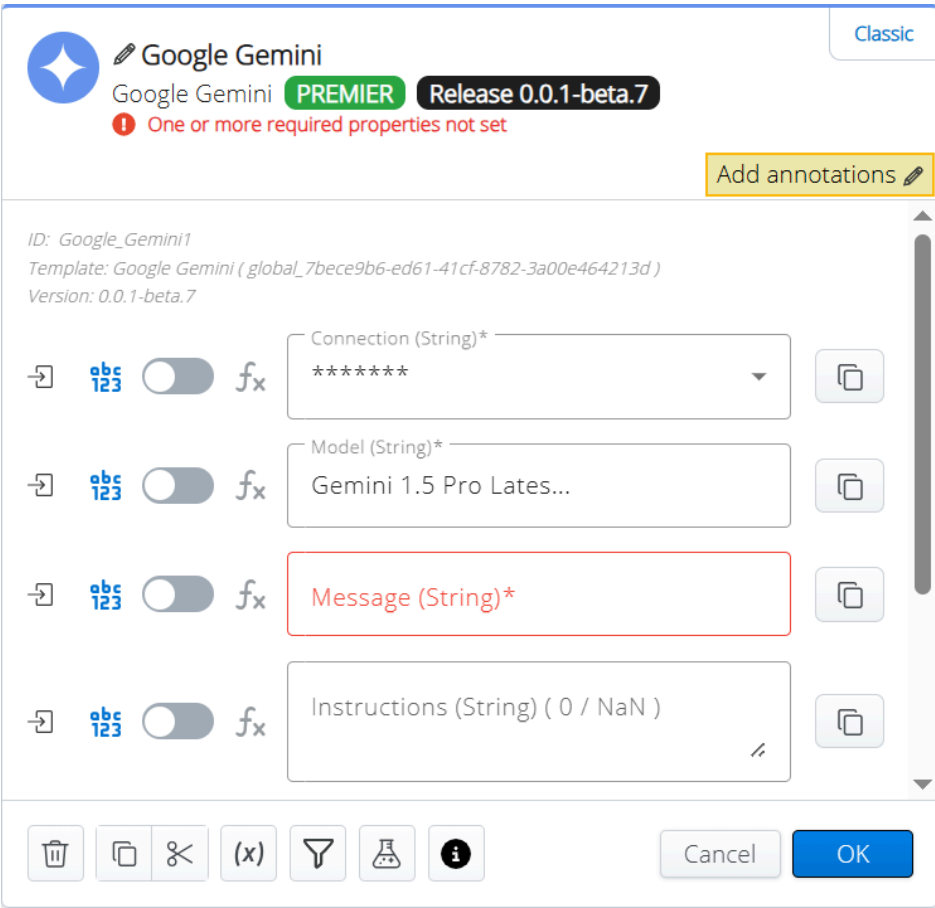


Figure 95: Google Gemini

- 7. In the **Google Gemini** activity screen, provide the configuration options as described in the Google Gemini - Field description table.

Note:
Mandatory fields are marked with an asterisk (*).

Table 297: Google Gemini – Field Description

Field	Description
Connection*	Select the connection created in the Google Gemini integration from the drop-down list.
Model*	Enter the AI model to use.
Message*	Enter the user message for sending to the AI model.
Instructions	Enter any specific instructions you want to provide to the AI model.

8. Click **OK** to add the Google Gemini activity to the flow.

9.6.3 Google Gemini Use Case

To use the Google Gemini integration within **Workflow Studio**, you can use either voice or chat media. The example below shows how to generate an AI response to a user when they send a chat message.

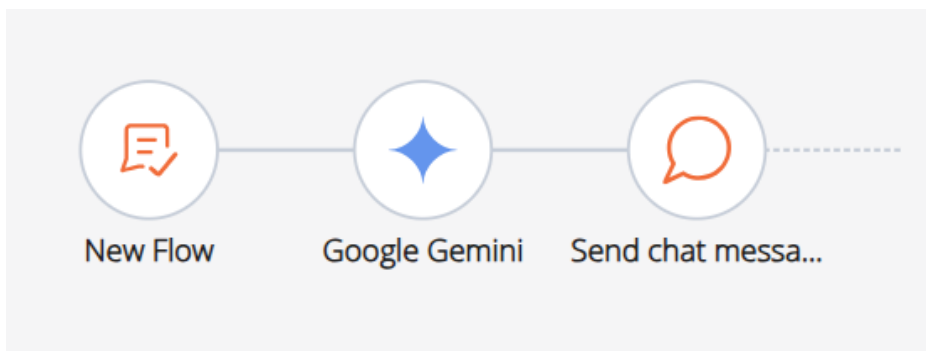


Figure 96: Google Gemini Use Case

To create the sample workflow for the above-mentioned use case, perform the following steps:

1. In the **Workflow Editor**, add a **Chat Message Received** trigger.
2. Add a **Google Gemini** activity to send the user's message to the AI model configured in the AI integration.
3. Add a **Send chat message** activity to send the response to the user from Google Gemini.

9.7 HTTP Integration (Premier)

HTTP Integration is required to use the HTTP Request activity in **Workflow Studio**.

9.7.1 Create HTTP Integration

To create an HTTP Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
- 3.



In the **Integrations Hub**, click icon on the **HTTP Integration** tile.

4. Click **Connect** to create a new connection.
5. Click **Next**.
6. Click **Add new** to add a connection using Basic authentication or API Key.

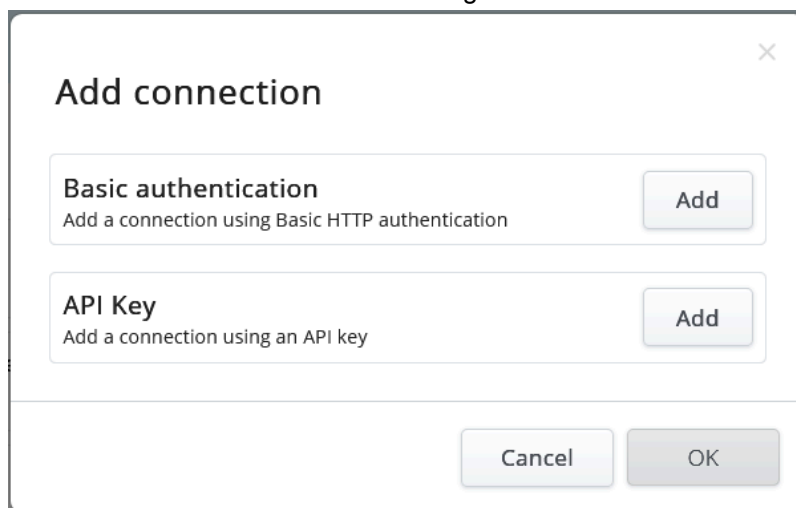


Figure 97: HTTP Integration - Add connection

7. In the **Add connection** screen, there are two options:
 - a. **Basic authentication:** A username and password are passed inside the header to be validated on the receiving end.
 - b. **API Key:** an API key is sent in the header, body, or query string as part of the request, which is validated on the receiving end.
8. Click **Add** next to the above option as required.
9. Click **OK**.

10. Click **Complete Setup** next to the selected connections.

The **Configure this connection** screen is displayed based on the selected options.

If the option is selected as **Basic Authentication**, the following screen is displayed.

Figure 98: Configure this connection - Basic Authentication

11. Provide the following details on the **Configure this connection – Basic Authentication** screen. For more information on fields, refer to the field description table.



Note:

Mandatory fields are marked with an asterisk (*).

Table 298: Basic Authentication – Field Description

Field	Description
Name	Enter the name of the connection used to identify the integration connection when creating the activity.
Username	Enter the username defined in the HTTP provider configuration.

Field	Description
Password	Enter the password defined in the HTTP provider configuration.

If the option is selected as **API Key**, the following screen is displayed.

The screenshot shows a modal window titled "Configure this connection" with a close button (X) in the top right corner. Inside the modal, there are four input fields, each with a label and an asterisk indicating it is required:

- Name ***: The input field contains the text "HTTP Integration API Key 3".
- Sent in (Enter body, header or url) ***: The input field contains the text "header".
- API Key name ***: The input field contains the text "Authorization".
- API Key value ***: The input field is empty.

Below the input fields, the word "Required" is written in red. At the bottom right of the modal, there is a blue button labeled "Create".

Figure 99: Configure this connection – API Key

12. Provide the following details on the **Configure this connection – API Key** screen. For more information on fields, refer to the field description table.

Note:
Mandatory fields are marked with an asterisk (*).

Table 299: API Key – Field Description

Field	Description
Name	Enter the name of the connection used to identify the integration connection when creating the activity.

Field	Description
Sent In (Enter body, header or url)	Enter one of three options in the text box: body, header, or url. The default is a header; depending on the external application, it may require the API Key to be passed in the header or URL.
API Key name	Enter the field name in the provider's application where the API Key resides; this is passed along with the API Key value. The default is Authorization .
API Key value	Enter the API Key , which can be found in the configuration of the external application.

13. Click **Create** once the required fields have been populated.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

14. Click **Continue**.

The newly created connection is displayed in the list.

9.7.2 Add an HTTP Request Activity

Once the HTTP connection is configured, you can use an HTTP Request activity within a flow.

To add HTTP Request activity in a flow, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create/Edit flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click on the flow editor to open the activity wheel.

- Under **Integration** activities, under the **Developer** category, select the **HTTP Request** activity to the flow.

The **HTTP Request Activity** popup screen is displayed.

Figure 100: HTTP Request Activity

- Select the **Authorization** Tab.
- In the **Integration** field, select the integration as **HTTP Integration** from the drop-down list.
- In the **Connection** field, select the required connection from the list.

This value is populated based on the integrations created under **HTTP integration** within this account.

- Click **OK**.

The **HTTP Request Activity** is added to the flow.

9.7.3 Business Use case

This topic provides a sample use case to illustrate how to create an HTTP integration with an external third-party application called Lemon Fox AI to translate a WAV file to text.

To create a new API Key, perform the following steps:

- Log into the **Lemonfox.ai** site.
- Click **APIs**.

The **AI API Documentation** page is displayed.

3. Click **Manage API Keys**.

The **Create a new API Key** popup screen is displayed.

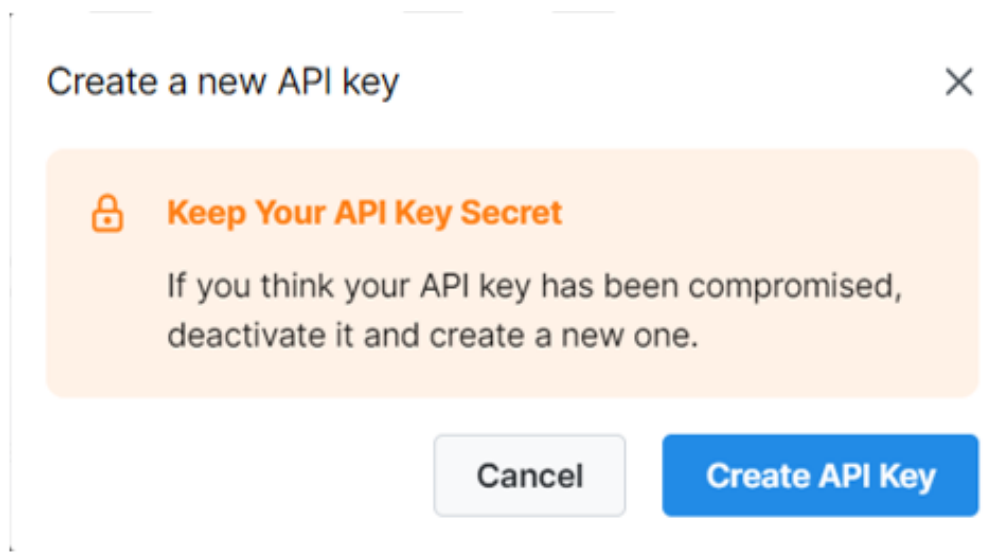


Figure 101: Create a new API key

4. Click **Create API Key**.

You will now see the API Key in the list.



5. Click the **Clipboard** icon to copy the key.

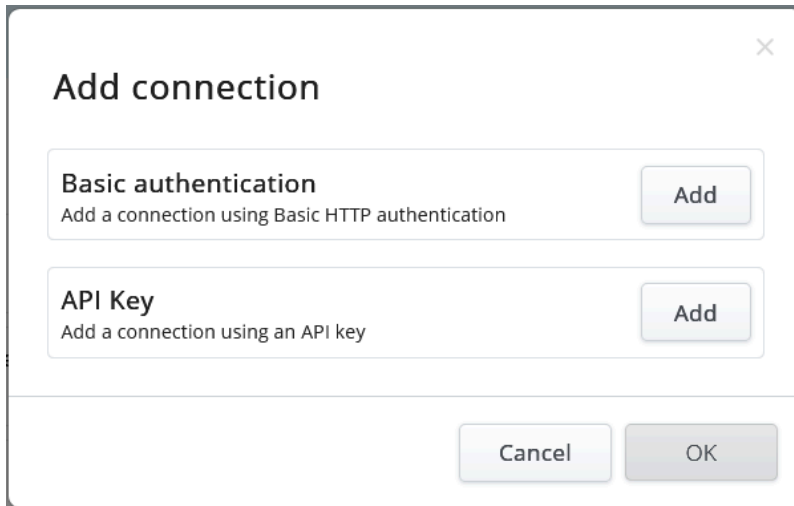
Note:
The key value is safe in case you need it.

To add a connection for the Lemon Fox AI integration, perform the following steps:

1. Log in to **Workflow Studio**.
2. On the **Workflow Studio** home page, under **Library**, click **Integrations**.

The **Integrations Hub** page is displayed.

3. In the **Integrations Hub**, click  icon on the **HTTP Integration** tile.

4. Click **Add New**.


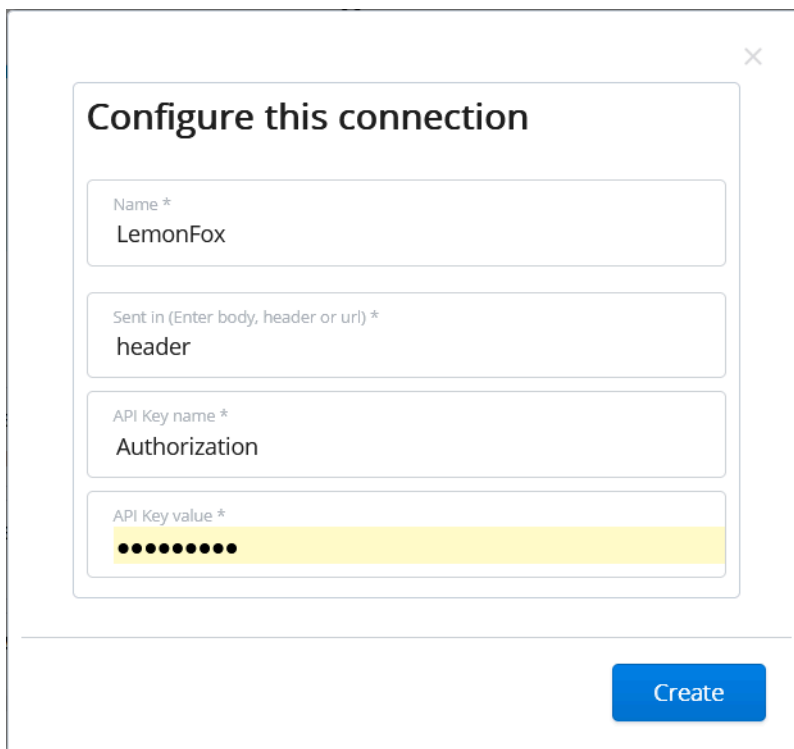
Add connection

Basic authentication
Add a connection using Basic HTTP authentication **Add**

API Key
Add a connection using an API key **Add**

Cancel **OK**

Figure 102: LemonFox AI - Add connection

5. Click **Add** next to the **API Key** option.6. Click **Complete Setup**.


Configure this connection

Name *
LemonFox

Sent in (Enter body, header or url) *
header

API Key name *
Authorization

API Key value *
●●●●●●●●

Create

Figure 103: LemonFox AI - Configure this connection

7. In the **Configure this connection** screen, provide the following details:

- a. In the **Name** field, enter a name for the connection as LemonFox.
- b. Keep the default values in the **Sent in** field.
- c. In the **API Key name** field, enter the value as **Authorization** while sending the API Key.
- d. In the **API Key value** field, paste the API Key value copied from the Lemonfox.ai application.

8. Click **Save**.

Now, the connection is complete and selectable within a flow.

To add an HTTP Request activity to a flow, perform the following steps:

1. Create a new flow with a trigger of your choice.

In this example, a Webhook trigger is used to accept an input of a filename.

The screenshot shows the configuration window for a 'Lemon Fox Example' Webhook trigger. The window has a 'Classic' tab in the top right. The title bar says 'Lemon Fox Example' with a pencil icon. Below the title, it says 'Webhook: Release 1.0.0' and 'Run on request'. A warning message states: 'It is recommended to set a Client Secret so external services may access this flow'. There is a yellow button labeled 'Add annotations' with a pencil icon. Below this, there is a text input field for 'ID' containing 'Webhook1'. Underneath, there is a section for 'FileName' with a 'String' type indicator. To the right of 'FileName' is a text input field containing 'https://www.r', a small icon with a square and an arrow, and a red 'X' button. Below the 'FileName' section are three buttons: '+ Input', '+ Throughput', and '+ Output'. At the bottom of the window, there is a toolbar with icons for delete, copy, paste, undo, and redo, followed by 'Cancel' and 'OK' buttons.

Figure 104: Lemon Fox - Webhook trigger

2. In a **Webhook** trigger, enter the file name input value as "https://www.mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/Samples/Goldwave/addf8-Alaw-GW.wav"
3. Add an **HTTP Request** action to use the Lemon Fox AI integration created.

4. In the **Body** tab, the JSON body definition should be provided, as shown in the figure below.

The screenshot shows the 'Http Request' configuration window. The 'Body' tab is selected, displaying a JSON body definition:

```

1 {
2   "file": "{{FileName}}",
3   "language": "english",
4   "response_format": "json"
5 }

```

The window includes a 'Test' button and a 'Cancel' button. The 'Advanced' tab is also visible.

Figure 105: LemonFox - HTTP Request

5. Provide the JSON definition with the following details:

- file: {{FileName}} – the value retrieved from the Webhook
- language: English
- response_format: json

6. In the **Authorization** tab, select the options for LemonFox integration as shown below.

The screenshot shows the 'Http Request' configuration window. The 'Authorization' tab is selected, displaying the following options:

- Integration:** HTTP Integration
- Connection:** LemonFox

The window includes a 'Test' button and a 'Cancel' button. The 'Advanced' tab is also visible.

Figure 106: Lemon Fox - HTTP Request Authorization

7. Click **OK**.

8. Run the flow in debug and click on the gumbdrop on the HTTP Request activity.
9. Click **Object** next to the **Body** field.

The **Export** dialog is displayed with the output.

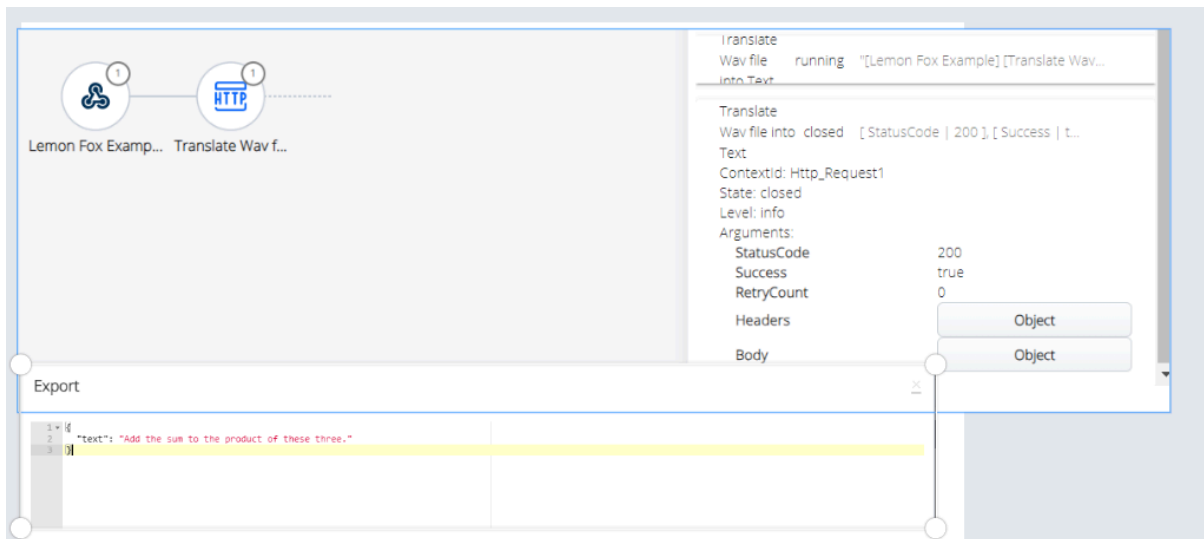


Figure 107: LemonFox - Export

9.8 Jira Cloud Integration (Premier)

Jira Cloud Integration allows you to create a connection to a JIRA Cloud service, which is required to use the Jira Cloud activities.

9.8.1 Create Jira Cloud Integration

You must obtain an API Token to integrate with Jira Cloud via Workflow Studio.

To create a Jira Cloud Integration in Workflow Studio, perform the following steps:

1. Login to **Workflow Studio**.
2. Under **Library**, click **Integrations**.
- 3.

In the **Integrations Hub**, click  icon on the **Jira Cloud** integration tile.

- Click **Add new** to create a new connection.

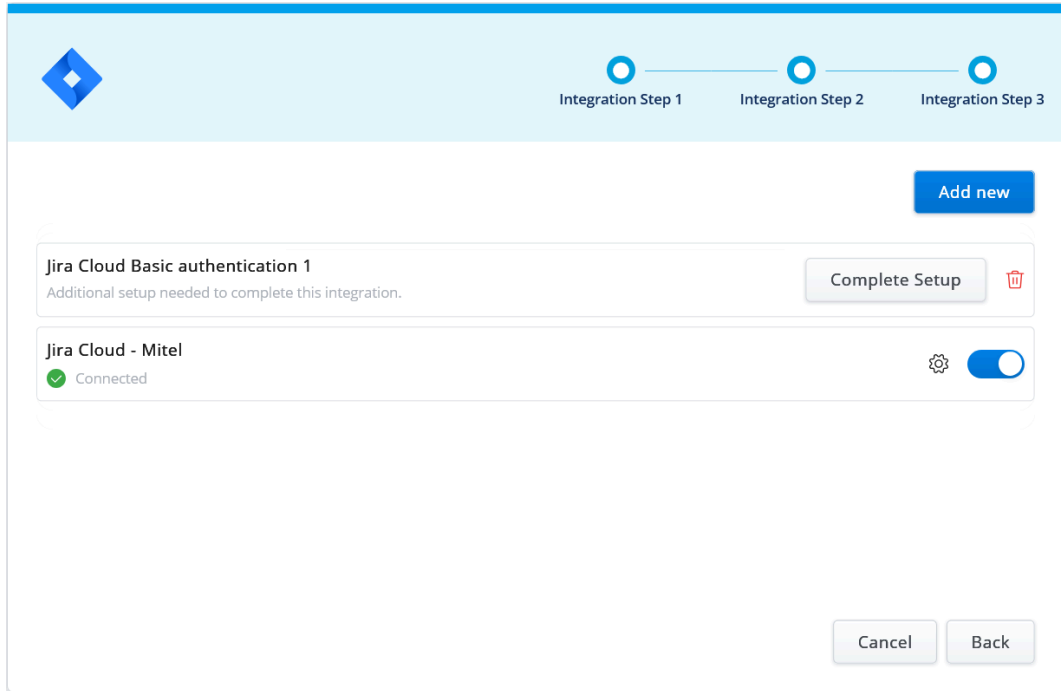


Figure 108: Jira Cloud Integration - Add new

- Click **Complete Setup**.

The screenshot shows a 'Configure this connection' dialog box. It has a title bar with a close button (X). The dialog contains four input fields, each with a red 'Required' label to its left. The first field is 'Name *' with the value 'Jira Cloud Basic authentication 1'. The second field is 'userid *'. The third field is 'API_TOKEN *'. The fourth field is 'domain *' with the value 'yourdomain.atlassian.net'. At the bottom right, there is a blue 'Create' button.

Figure 109: Jira Cloud Integration - Configure this connection

6. In the **Configure this connection** screen, provide the following details:

- a. Enter an appropriate connection name in the **Name** field.
- b. Enter your API Token in the **API Token** field.

Your API Token is copied from the Jira Cloud application.

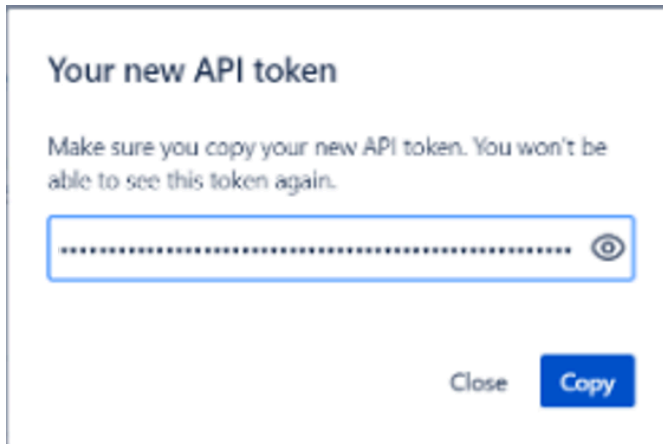


Figure 110: Jira Cloud - New API Token

Please do not share this token with anyone; it will allow access to your Jira Cloud APIs.

- c. Enter the appropriate domain in the **Domain** field.

7. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

8. Click **Continue**.

The newly created connection is displayed in the list. You can now use your Jira Cloud Integration in Workflow Studio.

9.8.2 Add Jira Cloud Activities

Once you have configured a connection, you can use the Jira Cloud connection to perform various actions in a workflow.

To configure the workflow, perform the following steps:

1. In the **Workflow Studio**, open the new or existing flow to which you want to add this activity.
2. In the **Workflow Canvas**, open the activity wheel and search for “JIRA” activities.

The list of JIRA activities are displayed.

Refer to the **Actions** section to view the definition of JIRA actions.


3. Click on any one of the actions to add it to the flow.
4. Configure the parameters in the respective fields.
5. Click **Save** to save the flow.

9.9 Microsoft Azure Integration (Premier)

Microsoft Azure Bot Integration allows you to create a connection to an Azure Bot Service, which is required when building out Azure AI bots to send messages from a third party like Workflow to Azure and vice versa.

9.9.1 Create Microsoft Azure Bot Integration

To create Microsoft Azure Bot Integration in Workflow Studio, perform the following steps:

1. Login to **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click  icon on the **Microsoft Azure Bot** integration tile.
4. Click **Add new** to create a new connection.

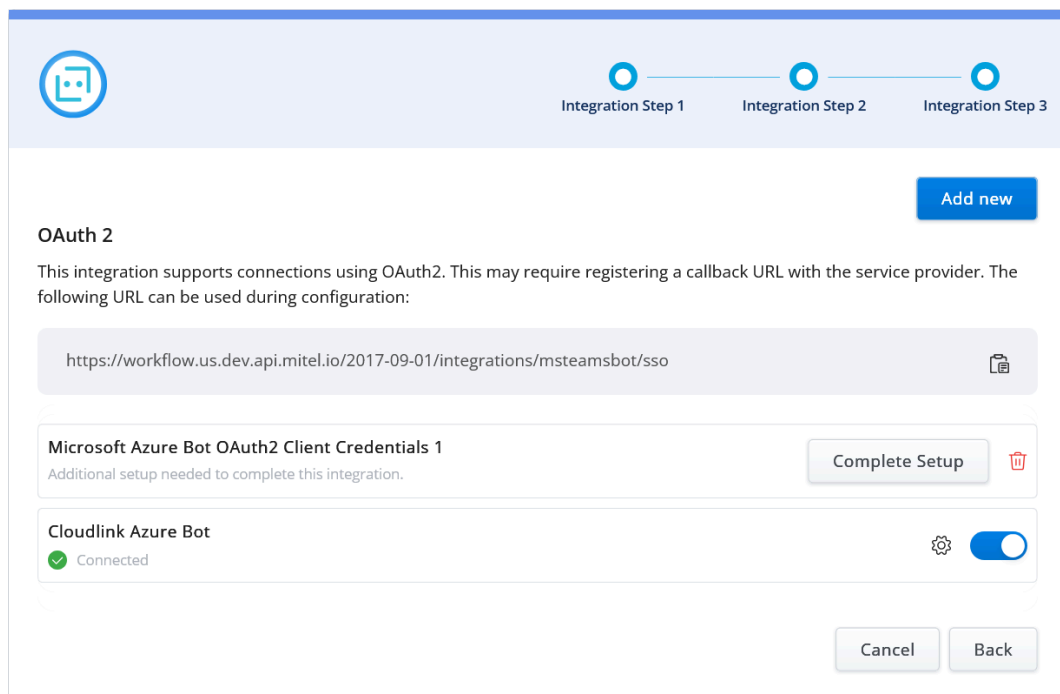


Figure 111: Microsoft Azure Integration - Add new

5. Click **Complete Setup**.

Figure 112: Microsoft Azure Integration - Configure this connection

6. In the **Configure this connection** screen, provide the following details:

- a. Enter an appropriate connection name in the **Name** field.
- b. Enter the client ID created within the Azure Portal in the **client_id** field.
- c. Enter the client secret created within the Azure Portal in the **client_secret** field.
- d. In the **Scope** field, the default scope is “{{client_id}}/.default”.
- e. In the **host** field, the default value is “smba.trafficmanager.net”.

7. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

8. Click **Continue**.

The newly created connection is displayed in the list. You can now use your Azure Integration in Workflow Studio.

9.10 Microsoft Graph Integration (Premier)


Microsoft Graph is a REST API built by Microsoft to allow access to Microsoft Office 365 Services like MS Teams and OneDrive. Workflow supports two types of authentications.

1. **OAuth2 Authorization Code** is a secure and commonly used authorization flow that allows applications to access resources on behalf of a user. The Graph API is often used in the context of Microsoft services like Microsoft Graph, which provides a unified endpoint to access data from various Microsoft services.
2. **OAuth2 Client Credentials** is typically used for server-to-server interactions, where an application needs to access resources without user interaction. This flow is suitable for applications that need to access resources that are not user-specific, such as accessing an API to read or write data.

9.10.1 Create Microsoft Graph Integration

To create Microsoft Graph Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
- 3.

In the **Integrations Hub**, click  icon on the **Microsoft Graph** integration tile.

4. Click **Add new** to create a new connection.

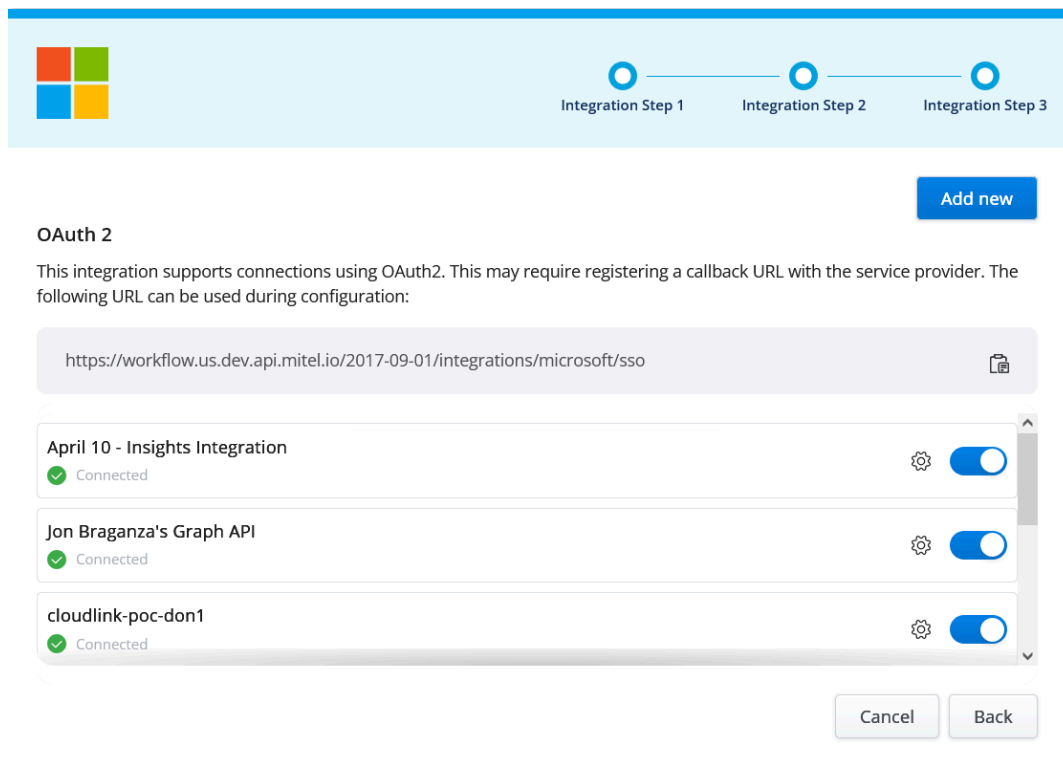
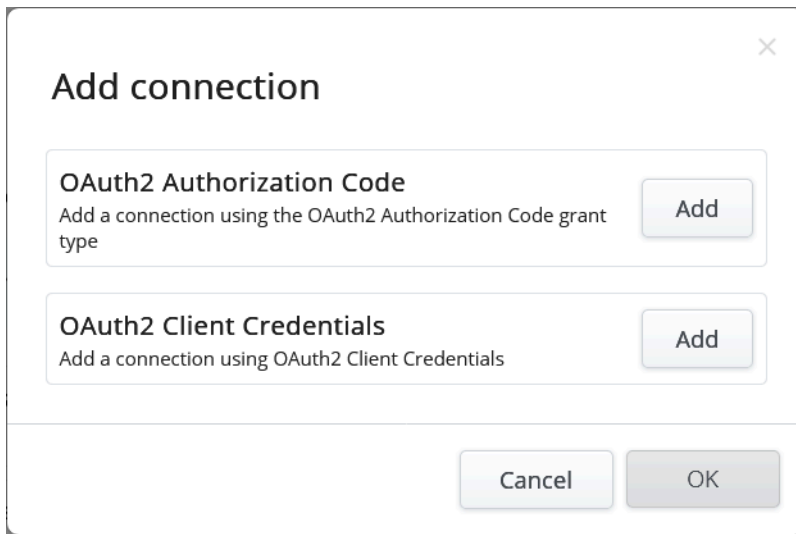


Figure 113: Graph Integration - Add new

5. Click **Add** to add a connection using OAuth2 Authorization Code or OAuth2 Client Credentials.



The image shows a dialog box titled "Add connection" with a close button (X) in the top right corner. Inside the dialog, there are two main sections. The first section is titled "OAuth2 Authorization Code" and contains the text "Add a connection using the OAuth2 Authorization Code grant type" followed by an "Add" button. The second section is titled "OAuth2 Client Credentials" and contains the text "Add a connection using OAuth2 Client Credentials" followed by an "Add" button. At the bottom of the dialog, there are two buttons: "Cancel" and "OK".

Figure 114: Graph Integration - Add connection

To add the connection using the OAuth2 Authorization Code, perform the following steps:

1. Click **Add** next to OAuth2 Authorization Code and click **OK**.

×

Configure this connection

Name *

Microsoft Graph API OAuth2 Authorization Code 1

tenant *

Required

client_id *

Required

scope *

openid email profile offline_access

prompt

consent

domain_hint

organizations

client_secret *

Required

Create

Figure 115: Graph Integration - Configure this connection

2. In the **Configure this connection** screen, provide the following details. For more information on fields, refer to the field description table.



Note:

Mandatory fields are marked with an asterisk (*).

Table 300: OAuth2 Authorization Code – Field Description

Field	Description
Name	Enter the name of the connection.

Field	Description
Tenant	Enter the application's object ID.
client_Id	Enter the application's client ID.
Scope	Enter the list of scopes in a space-separated format to indicate the resources and operations the application requests.
client_secret	Enter the application's client secret code.

3. Click **Create**

The **Microsoft Login** page is displayed to authenticate the entered email address and password.

4. Click **Continue.**

The newly created connection is displayed in the list.

To add the connection using the OAuth2 Client Credentials, perform the following steps:

1. Click **Add** next to OAuth2 Client Credentials and click **OK**.

×

Configure this connection

Name *

Microsoft Graph API OAuth2 Client Credentials 1

tenant *

Required

client_id *

Required

client_secret *

Required

scope *

https://graph.microsoft.com/.default

Create

Figure 116: Graph Integration - Configure this connection

2. In the **Configure this connection** screen, provide the following details. For more information on fields, refer to the field description table.

Note:
Mandatory fields are marked with an asterisk (*).

Table 301: OAuth2 Client Credentials – Field Description

Field	Description
Name	Enter the name of the connection.
Tenant	Enter the application's object ID.

Field	Description
client_id	Enter the application's client ID.
client_secret	Enter the application's client secret code.
scope	Enter the list of scopes in a space-separated format to indicate the resources and operations the application requests.

3. Click **Create**.

The connection is validated against the Microsoft authorization URL (<https://login.microsoftonline.com/{{tenant}}/oauth2/v2.0/token>).

Once the connection is validated, the status is presented to the user.

9.10.2 Add a Graph Connection activity

Once the Microsoft Graph API connection is configured, you can use the connection within flows/actions/triggers to make Graph API calls or use Graph-specific activities.

There are two different ways to add the Graph activity.

- Using HTTP activity to make Graph API calls

Http Request Release 1.0.0 Advanced

Http Request Release 1.0.0
Integrate with RESTful Web APIs

Add annotations

GET Test

Headers Body Authorization Testing

Integration

Connection

Cancel OK

Figure 117: HTTP Request

- Use in-built Microsoft Graph activities



Figure 118: Search Graph Activities

Refer to the **Activities** section to view a description of all supported Microsoft activities.

9.10.3 Retrieve the Teams ID

You must know the Teams ID while configuring activities such as 'Send message to MS Teams channel'.

To retrieve the Teams ID for a particular team using Microsoft Teams, perform the following steps:

1. Log into **Microsoft Teams** application.
2. Click on the **Teams** icon in the left navigation menu.
3. Right-click on the team name for which you want to find the Team ID.
4. Select **Get link to team**.

The URL link is displayed with Team ID next to the `groupId` parameter, as shown in the below example:

```
https://teams.microsoft.com/l/team/19%3A1c3101a2a5264d248ff46d3193a1a05a%40thread.tacv2/conversations?groupId=5db8a0ea-6460-4ff3-9c45-a297a5f5d0ef&tenantId=123c8de3-bf07-4569-b282-b7512a1c9999
```

To retrieve the Teams ID using Microsoft Graph API in Graph Explorer, perform the following steps:

1. Open Graph Explorer.
2. Sign in with your user credentials, which are used for signing in to Microsoft Teams.
3. Fill in the required details as mentioned below.

```

HTTP Method : GET
Version : v1.0
URL : https://graph.microsoft.com/v1.0/groups?$filter=displayName eq
      '{teams-display-name}'&$select=id

```

4. Click Run query.

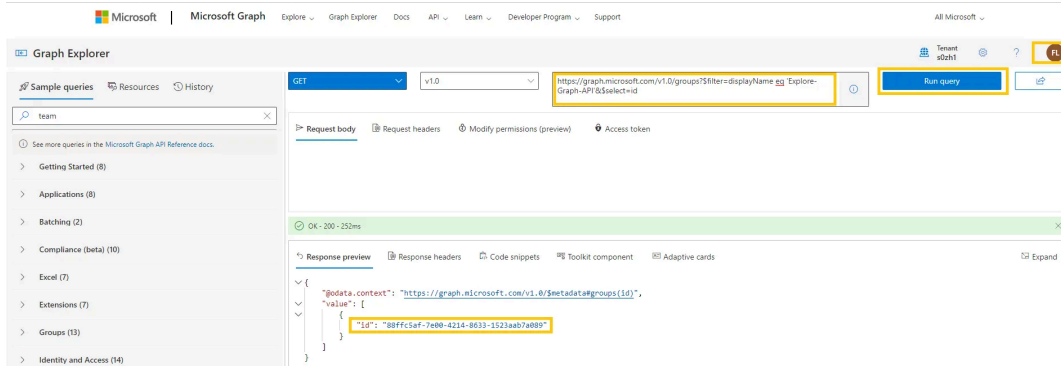


Figure 119: Graph Explorer

Note:

In case of a Forbidden error thrown by Graph API, find the missing permission from the error message. In this case, Team.ReadBasic.All permission is displayed.

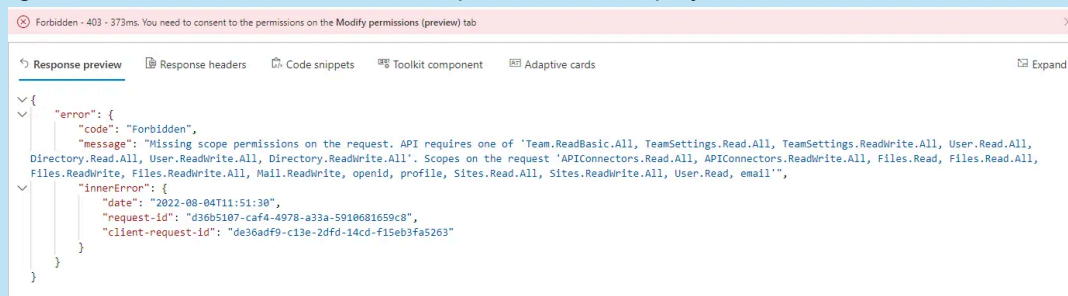


Figure 120: Forbidden Error

To add the missing permission, perform the following steps:

1. Navigate to **Modify permissions (preview)** tab.

Request body Request headers **Modify permissions (preview)** Access token

Permissions (7)
One of the following permissions is required to run the query. Select a permission and click Consent.

Permission	Description	Admin consent required	Status
Directory.Read.All	Allows the app to read data in your organization's directory.	✓	Consent
Directory.ReadWrite.All	Allows the app to read and write data in your organization's directory, such as other users, groups. It does not allow the app to delete users or groups, or reset user passwords.	✓	Consent
Team.ReadBasic.All	Read the names and descriptions of teams, on your behalf.	*	Consented
TeamSettings.Read.All	Read all teams' settings, on your behalf.	✓	Consent
TeamSettings.ReadWrite.All	Read and change all teams' settings, on your behalf.	✓	Consent

Figure 121: Modify permissions (preview)

2. To provide the permission, click the **Consent** button next to the permission as you prefer.

9.10.4 Permissions Required

To use the Microsoft Graph activities in Workflow Studio, the following permissions are required in the Azure portal based on the application used.

Note:

When changes are made to permissions, they may not occur immediately. If the function still fails after the changes are saved, recheck the permissions and then retest.

While configuring Graph Integration using **OAuth2 Authorization Code** connection type, the scope must be added. To inherit the default scopes from the application that is registered in Azure, enter the value in the scope field as `https://graph.microsoft.com/.default offline_access`.

Note:

`offline_access` is required while using the default scope. Otherwise, the connection will be invalid when the workflow refreshes the token.

9.10.4.1 Permissions for Excel Actions

To use the Excel actions to retrieve the worksheets and enable the Workflow Studio to read/update/add/delete records in a table within the worksheet, the following permissions are required in the **Application Registration** in the Azure portal. The Excel actions are as follows:

- Excel get table rows
- Excel add, update or delete table rows

Delegated Permission (OAuth2 Authorization Code)

Table 302: Delegated Permission

Name	Type
Files.ReadWrite.All	Delegate
User.ReadBasic.All	Delegate
User.Read	Delegate

Application Permission (OAuth2 Client Credentials)

This permission is not available as it would be a security concern to have Excel worksheet access for all users in an organization.

Scope

The following scope must be configured inside the **Connection** under **Scopes**.

```
openid email profile offline_access Files.ReadWrite.All
```

9.10.4.2 Permissions for One Drive Actions

To use the One Drive actions in Workflow Studio, the following permissions are required in the **Application Registration** within the Azure portal. The One Drive actions are as follows:

- OneDrive - File upload
- OneDrive - Get file contents

Delegated Permission (OAuth2 Authorization Code)

Retrieve and upload the files to a OneDrive location.

Table 303: Delegated Permission

Name	Type
Files.ReadWrite.All	Delegate
User.ReadBasic.All	Delegate
User.Read	Delegate

Application Permission (OAuth2 Client Credentials)

This permission is not available as it would be a security concern to have One Drive files/folders access for all users.

Scope

The following scope must be configured inside the **Connection** under **Scopes**.

```
openid email profile offline_access Files.ReadWrite.All
```

9.10.4.3 Permissions for Teams Find User Action

To use the **Teams find user** action in Workflow Studio, the following permissions are required in the **Application Registration** within the Azure portal.

Delegated Permission (OAuth2 Authorization Code)

Retrieves user data based on filter parameters.

Table 304: Delegated Permission

Name	Type
User.Read	Delegate
User.Read.All	Delegate

Application Permission (OAuth2 Client Credentials)

Retrieves the users by filtering on any property in the User object.

Table 305: Application Permission

Name	Type
User.Read	Delegate
User.Read.All	Application

Scopes

The following scope must be configured inside the **Connection** under **Scopes**.

```
openid email profile offline_access
```

9.10.4.4 Permissions for Teams Calendar Actions

To use the **Teams Calendar** actions in Workflow Studio, the following permissions are required in the **Application Registration** within the Azure portal. The Teams Calendar actions are as follows:

- Get calendar availability
- Set calendar event

Delegated Permission (OAuth2 Authorization Code)

Get and Set Calendar for the connected user.

Table 306: Delegated Permission

Name	Type
Calendars.ReadWrite	Delegate
User.Read	Delegate

Application Permission (OAuth2 Client Credentials)

Get and Set calendar against any user.

Table 307: Application Permission

Name	Type
Calendars.ReadWrite	Application
User.ReadAll	Application

Scope

The following scope must be configured inside the **Connection** under **Scopes**.

```
openid email profile offline_access Calendars.ReadWrite
OnlineMeetings.ReadWrite
```

9.10.4.5 Permissions for Teams Send Message to Channel

This activity provides the ability to send a message to MS Teams channel. To use the **Send message to MS Teams channel** action in Workflow Studio, the following permissions are required in the **Application Registration** within the Azure portal.

Delegated Permission (OAuth2 Authorization Code)

Send a message to a channel.

Table 308: Delegated Permission

Name	Type
User.Read	Delegate
ChannelMessage.Send	Delegate

Application Permission (OAuth2 Client Credentials)

Find the channelId based on the the ChannelName and TeamsID provided.

Table 309: Application Permission

Name	Type
User.Read	Delegate
Channel.ReadBasic.All	Application
ChannelSettings.Read.All	Application
Teamwork.Migrate.All	Application

Scopes

The following scope must be configured inside the **Connection** under **Scopes**.

`openid email profile offline_access channelmessage.send`

9.10.4.6 Permissions for Teams Presence

To use the **Teams presence** action in Workflow Studio, the following permissions are required in the **Application Registration** within the Azure portal.

Delegated Permission (OAuth2 Authorization Code)

Get/Set presence for the user that is authenticated in the connection.

Table 310: Delegated Permission

Name	Type
Presence.Read	Delegate
Presence.ReadWrite	Delegate
User.Read	Delegate

Application Permission (OAuth2 Client Credentials)

Get/Set presence for any user.

Table 311: Application Permission

Name	Type
Presence.Read	Application
Presence.ReadWrite	Application
User.ReadAll	Application

Scopes

The following scope must be configured inside the **Connection** under **Scopes**.

```
openid email profile offline_access Presence.Read Presence.Read.All
Files.ReadWrite.All
```

9.11 OpenAI Assistants Integration (Premier)

Open AI Assistants Integration is required to use the ChatGPT and OpenAI assistants activities in **Workflow Studio**.

9.11.1 Prerequisites

Before starting the integration, the following accounts are needed.

- OpenAI account is required.

For more details on account setup, refer to www.openai.com.

- Mitel Workflow Studio account is also required.
- Please contact your channel partner for access to CloudLink Services.

9.11.2 Set up your OpenAI Assistants

This topic provides step-by-step instructions on how to set up your OpenAI Assistant. There are many configurations, and you should choose the best options for your requirements.

1. Launch the OpenAI Website. (<https://platform.openai.com/docs/overview>)
2. On the left pane, click the **Assistants** tab.
3. Click **Create** to create a new assistant.

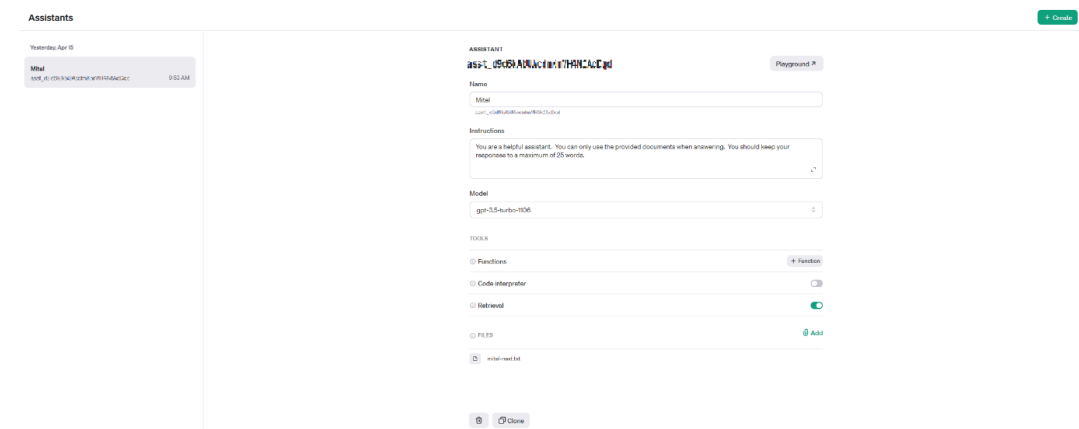


Figure 122: Create Assistant



Note:
Refer to the **OpenAI documentation** for more details.

4. On the **Create Assistant** screen, enter the following details in the respective fields. For more information, refer to the field description table.

Table 312: Create Assistant – Field Description

Field	Description
Name	Enter the name of your assistant.
Instructions	Enter the instructions for your assistant.
Model	Enter the GPT model for your assistant.

Field	Description
Retrieval	Select the toggle ON to allow your assistant to look up the reference material provided. Only certain GPTs support this function.
Files	Click Add to upload any files required for your assistant.

5. Under the **Settings** menu, click **Billing** to set up your billing details.

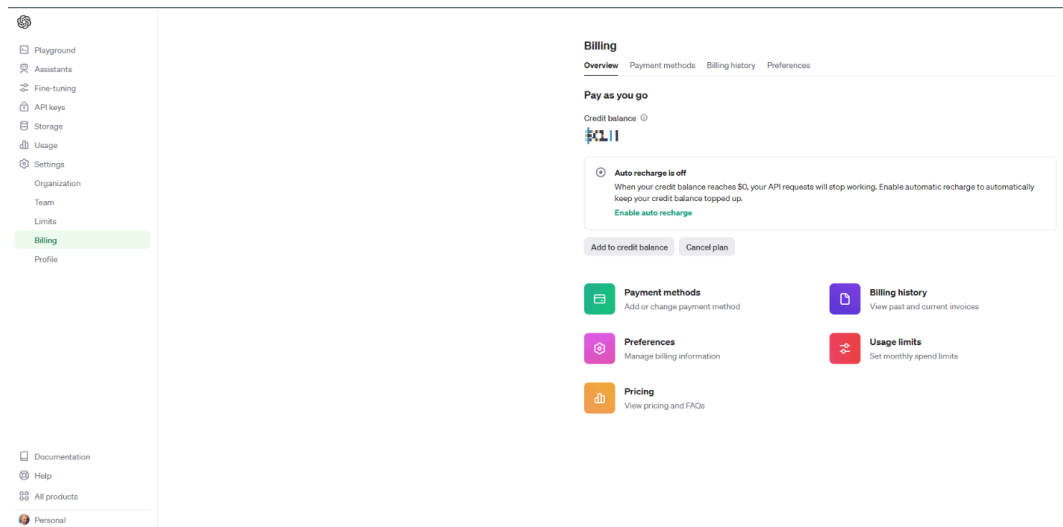


Figure 123: Billing

9.11.3 Create OpenAI Integration

To create Outlook Integration in Workflow Studio, perform the following steps:

1. Login to **Workflow Studio**.
2. Under **Library**, click **Integrations**.
- 3.

In the **Integrations Hub**, click  icon on the **OpenAI** integration tile.

4. Click **Add new** to create a new connection.

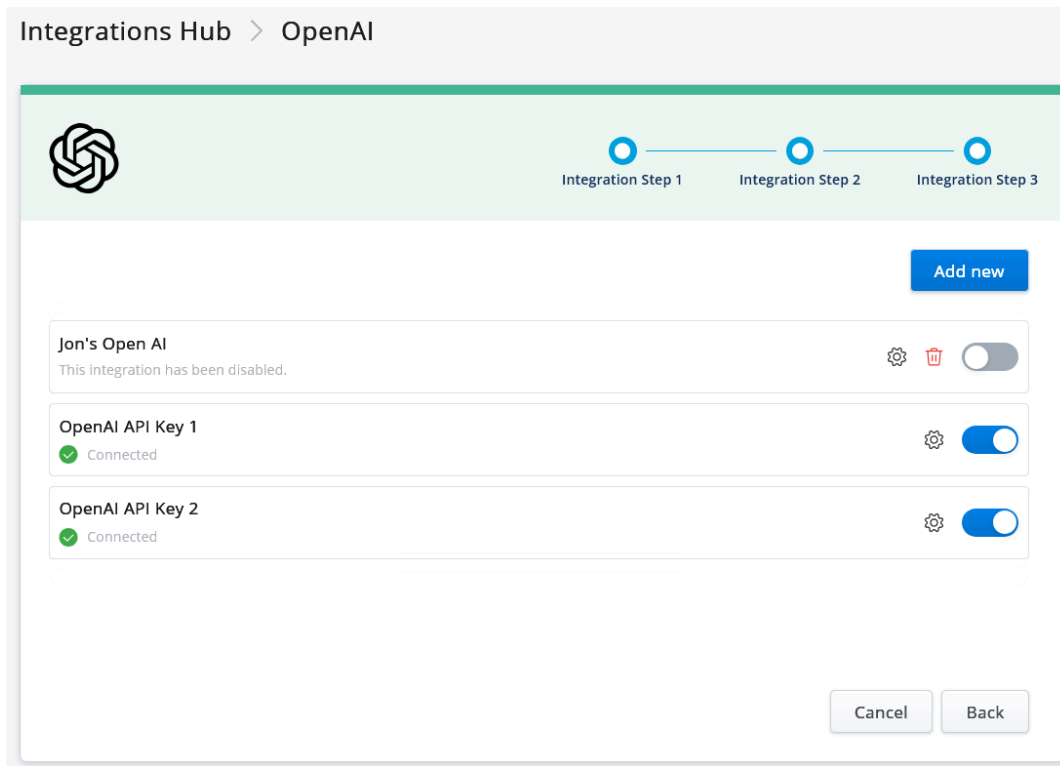


Figure 124: OpenAI Integration - Add new

5. Click **Complete Setup**.

The screenshot shows a 'Configure this connection' dialog box. It has a title bar with a close button (X). The main content area has two input fields: 'Name *' with the value 'OpenAI API Key 3' and 'OpenAI API Key *'. Below the second field, the word 'Required' is written in red. At the bottom right, there is a blue 'Create' button.

Figure 125: OpenAI Integration - Configure this connection

6. In the **Configure this connection** screen, provide the following details:

- a. Enter an appropriate connection name in the **Name** field.
- b. Enter your API Key in the **OpenAI API Key** field.

Your API Key can be found under the **API Keys** section of the OpenAI application.

Please do not share this key with anyone; it will allow access to your OpenAI APIs.

7. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

8. Click **Continue**.

The newly created connection is displayed in the list. You can now use your OpenAI Integration in Workflow Studio.

9.11.4 Configure the media types

Here, you can find details on how to configure different media types with OpenAI Assistants. Before proceeding, a working integration with the specified media type is required.

9.11.4.1 Voice Media

To configure Voice Media with Voice Assist, you will require a configured flow connected to the relevant voice media controllers, as specified in the Voice Assist documentation.

1. On the **Workflow Studio** home page, click **Create flow**.
2. In the **Workflow Canvas**, search and select an **Endpoint Ringing** trigger.
3. Click **Add**.
4. Select a valid phone number from the list.

5. Add an **Open AI Assistants** activity and configure it with the required settings:

OpenAI Assistants Classic

OpenAI Assistants **Release 0.0.1-beta.5**
Uses the Open AI GPT Assistants API to retrieve responses.

[Add annotations](#)

ID: OpenAI_Assistants1
Template: OpenAI Assistants (global_e5366586-bb92-4081-8f88-0c683a464e98)
Version: 0.0.1-beta.5

Connection (String)* *****

Assistant (String)* Mitel

Message (String)
Dial_Menu.Collector...

Instructions (String) (0 / NaN)

ThreadId (String)

Response
OpenAI_Assistan...
String

Success
OpenAI_Assistan...
Boolean

Output
OpenAI_Assistan...
Object

Cancel OK

Figure 126: OpenAI Assistants

6. Click **OK** to complete the Voice Media activity configuration with OpenAI Assistant.

9.11.4.2 Chat Media

To configure a chatbot using Open AI Assistants, you should choose a Chat Message received trigger and configure your assistant to answer the message based on the programming.

1. On the **Workflow Studio** home page, click **Create flow**.
2. In the **Workflow Canvas**, search and select a **Chat message received** trigger.
3. Click **Add**.

4. Add an **OpenAI Assistants** activity and configure it with the required settings:

OpenAI Assistants Classic

OpenAI Assistants **Release 0.0.1-beta.5**
Uses the Open AI GPT Assistants API to retrieve responses.

[Add annotations](#)

ID: OpenAI_Assistants1
Template: OpenAI Assistants (global_e5366586-bb92-4081-8f88-0c683a464e98)
Version: 0.0.1-beta.5

Connection (String)*

Assistant (String)*
Mitel

Message (String)
Body

Instructions (String) (84 / NaN)
You are a helpful assistant. You will tell people about Mitel products and services.

ThreadId (String)

Response
OpenAI_Assistan...
String

Success
OpenAI_Assistan...
Boolean

Output
OpenAI_Assistan...
Object

Cancel OK

Figure 127: OpenAI Assistants

5. Click **OK** to complete the Chat Media activity configuration with OpenAI Assistant.

9.12 Outlook Integration (Essential)

Outlook Integration is required to use the Email activity in **Workflow Studio**. Adding Outlook Integration to a customer account allows the Mitel Partner or Account Admin to send an email through Office 365 to selected email addresses.

9.12.1 Create a new application/registration within Microsoft Azure (Entra)

To create a new application/registration with Microsoft Azure (Entra), perform the following steps:

1. Log into the Azure portal (<https://portal.azure.com/>).
2. Navigate to **Enterprise Applications/App registrations**.
3. Create a **New application/New registration**.

**Note:**

You must select **Create your own application** from the **Entra Gallery** page for new applications.

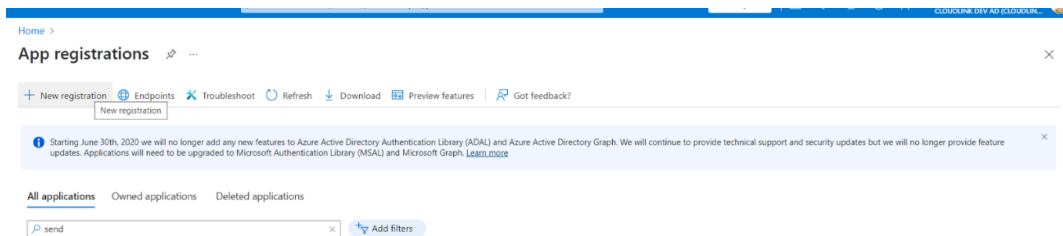


Figure 128: App Registrations

4. Enter a unique name for the **New application/New registration**.

**Note:**

This app will use the email address of the user who signs in. It is recommended that you include the email address in the Application/Registration name to allow you to identify the user for each connection easily. It is also recommended that you use the same name in the **Name** field while creating the connection in Workflows.

5. Add the permissions as described below:

+ Add a permission		✓ Grant admin consent for CloudLink Dev AD		
API / Permissions name	Type	Description	Admin consent requ...	Status
Microsoft Graph (2)				
Mail.Send	Delegated	Send mail as a user	No	✓ Granted for CloudLink D... ***
User.Read.All	Delegated	Read all users' full profiles	Yes	✓ Granted for CloudLink D... ***

Figure 129: Add Permissions

**Note:**

Depending on your view, you must navigate to **Application Registration** before you can add your permissions.

Once completed, you need to grant admin consent for MitelApps UK.

6. Create a Web Redirect URL - <https://workflow.us.api.mitel.io/2017-09-01/integrations/office365outlook/ssso>.



Note:

You may need to replace the region in this example <https://workflow.us>. with “eu” or “ap” based on where your CloudLink account is hosted.

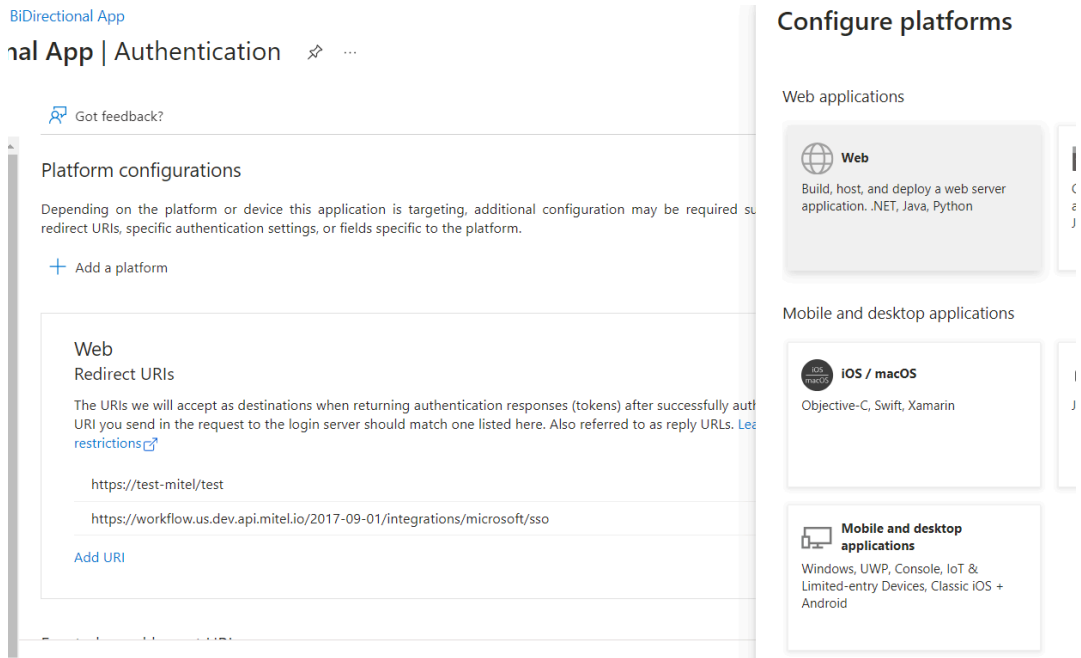


Figure 130: Platform Configurations

7. Select the **Client Secrets** tab on the **Certificates & secrets** page and click a client secret.

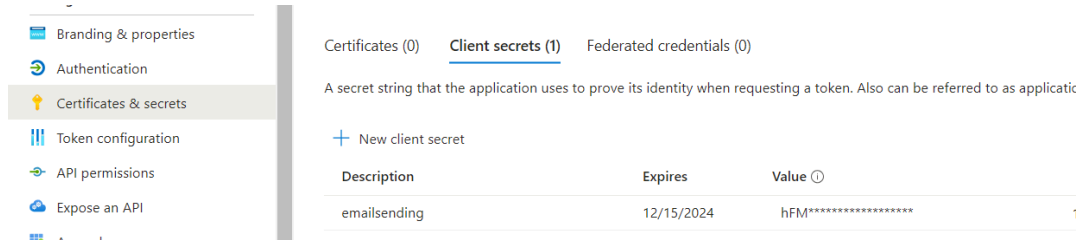


Figure 131: Certificates and secrets - Client secrets




Note:

If you navigate away from the **Certificates & secrets** page, the secret is hidden, which will be required when creating the Workflow integration. In this scenario, it is good practice to stay on this page while the Workflow integration is completed.

9.12.2 Create Outlook Integration

To create Outlook Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click  icon on the **Microsoft Office 365 – Outlook** integration tile.
4. Click **Add new** to create a new connection.
5. Click **Complete Setup**.
6. Click **Add** under the **OAuth2 Authorization Code**.

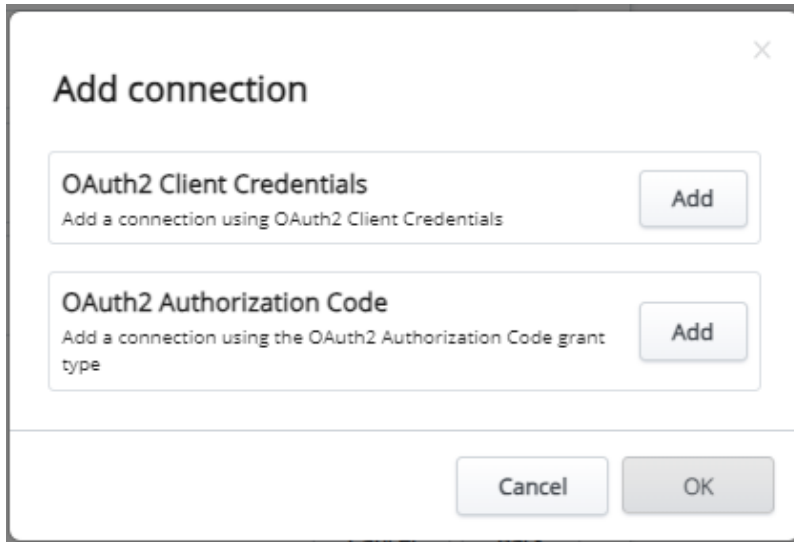


Figure 132: Outlook Integration - Add connection

7. Click **OK**.

8. Click **Complete Setup**.

The screenshot shows a 'Configure this connection' dialog box with a close button (X) in the top right corner. The dialog contains several input fields with labels and asterisks indicating required fields:

- Name ***: Microsoft Office 365 - Outlook OAuth2 Authorization Code 1
- Tenant Id ***: (Empty field)
- Application (Client) Id ***: (Empty field)
- Scope ***: openid email profile offline_access mail.send User.ReadBasic.A
- Prompt**: consent
- Domain Hint**: organizations
- Client Secret (Value) ***: (Empty field)

Red text 'Required' appears below the 'Tenant Id', 'Application (Client) Id', and 'Client Secret (Value)' fields. A blue 'Create' button is located at the bottom right of the dialog.

Figure 133: Outlook Integration - Configure this connection

9. In the **Configure this connection** screen, enter the respective fields.

The screenshot shows the 'Outlook Application Details' screen with a section titled 'Essentials' (indicated by an upward arrow). The fields are organized into two columns:

Field	Value	Field	Value
Display name	: CAP - Email sending - Adam	Client credentials	: 0 certificate
Application (client) ID	: 818e75-4b2-4c8-2d41-fa5b3c3310	Redirect URIs	: 1 web, 0 sp
Object ID	: e	Application ID URI	: Add an App
Directory (tenant) ID	: e	Managed application in l...	: CAP - Email
Supported account types	: My organization only		

Figure 134: Outlook Application Details

The best practice is to include the email address with the name to indicate the user using the integration.

10. Click Create.

You will be redirected to a login screen.



- a. Enter the email address and password of the user from whom the emails will be sent.

Once completed, the connection will be listed with a green indicator showing “it’s connected.”



9.12.3 Add an Office 365 Outlook - Send Email activity

To add an Office 365 Outlook – Send Email activity in a flow, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create flow**.
2. Search and select **Webhook** in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click  icon to add the new activity to the flow.
6. Under **Integration** activities, click  icon to add Outlook Activity to the flow.

The **Email Header** tab is displayed.

Figure 135: Email Header

7. On the **Email Header** tab, select the following details.

- a. In the **Connection (String)** field, select the integration you created from the drop-down list.
- b. In the **From (String)** field, select the user from whom the email must be sent.

The list displays the users based on the connection selected.

This field will be disabled, if the Office 365 Outlook User connection uses the "OAuth2 Authorization Code," which can only be sent by the authorized user.

- c. In the **Reply to email** field, enter the email address to which the recipient replies.

8. Click the **Email Content** tab to populate the **Subject** and **Body** of the email you want to send.

The **Email Content** tab is displayed.

Figure 136: Email Content

9. On the **Email Content** tab, enter the following details.

- a. In the **Email Subject** field, enter the subject of the email response.

Note:

You can reference workflow variables.

For example, In the **Email Subject** box, you can use the following: Message is from {{CallerName}}.

This would populate the subject line with "Message is from" and the caller's name collected from the workflow.

- b. In the **Email Content** text box, enter the body of the response message.

10. Click the **Recipients** tab to populate the email addresses to which the email will be sent.

The **Recipients** tab is displayed.

The screenshot shows the 'Office 365 Outlook - Send Email' configuration window. The 'Recipients' tab is selected, showing fields for 'Add recipients', 'Carbon copy' (Type CC recipients), and 'Blind Carbon copy' (Type BCC recipients). The 'Advanced' tab is visible in the top right corner. A yellow 'Add annotations' button is also present. At the bottom, there are icons for trash, copy, paste, undo, redo, and a testing icon (flask), along with 'Cancel' and 'OK' buttons.

Figure 137: Recipients

11. On the **Recipients** tab, enter the recipient's email address in the respective fields.
12. Click the **Attachments** tab to drag and drop or select a file from their local computer to include in the email.

The **Attachments** tab is displayed.

The screenshot shows the 'Office 365 Outlook - Send Email' configuration window with the 'Attachments' tab selected. A dashed box contains the text 'Drag and Drop or Select a File' and '0 bytes of 4MB selected'. The 'Advanced' tab is visible in the top right corner. A yellow 'Add annotations' button is also present. At the bottom, there are icons for trash, copy, paste, undo, redo, and a testing icon (flask), along with 'Cancel' and 'OK' buttons.

Figure 138: Attachments

13. Click the **Testing** (🧪) icon at the bottom of the activity box to test the activity.

**Note:**

Make sure the mandatory fields (marked with an asterisk and highlighted in red) are populated.

14. Click **Run Test** to run the test.

The results are displayed in the Test area.

The screenshot shows the 'Office 365 Outlook - Send Email - Testing' interface. On the left, there are input fields for 'Connection (String)*' (MICC VOiceMail User), 'To (Array)' (Items (1)), 'Cc (Array)' (Items (0)), 'Bcc (Array)' (Items (0)), 'Subject (String)*' (test), and 'ContentType (String)*' (html). Each field has a toggle switch and a formula icon. On the right, there are buttons for 'View RAW' and 'Run Test Again'. Below these buttons, the test results are displayed: 'ResponseBody (Object)' (Object {}), 'StatusCode (Number)' (202), and 'Success (Boolean)' (true).

Figure 139: Test Results

15. Click **OK** to add the Outlook activity to the flow.

9.13 Salesforce Integration (Premier)

Salesforce Integration is required to use the salesforce-related activities in **Workflow Studio**.

9.13.1 Setup an OAuth flow

You must obtain an API Token and set up an OAuth flow to integrate with Salesforce via Workflow Studio. Perform the following steps:

1. Log into your Salesforce domain.

2. Navigate to App Manager.

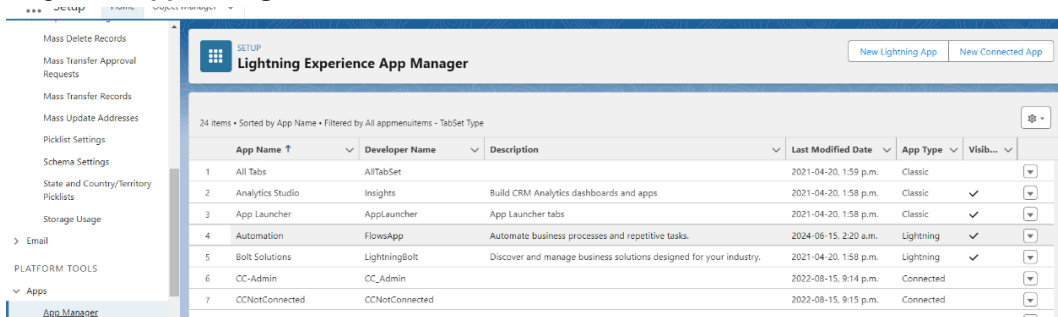


Figure 140: App Manager

3. Click **New Connected App**.

4. Provide the details in the respective fields as shown below.

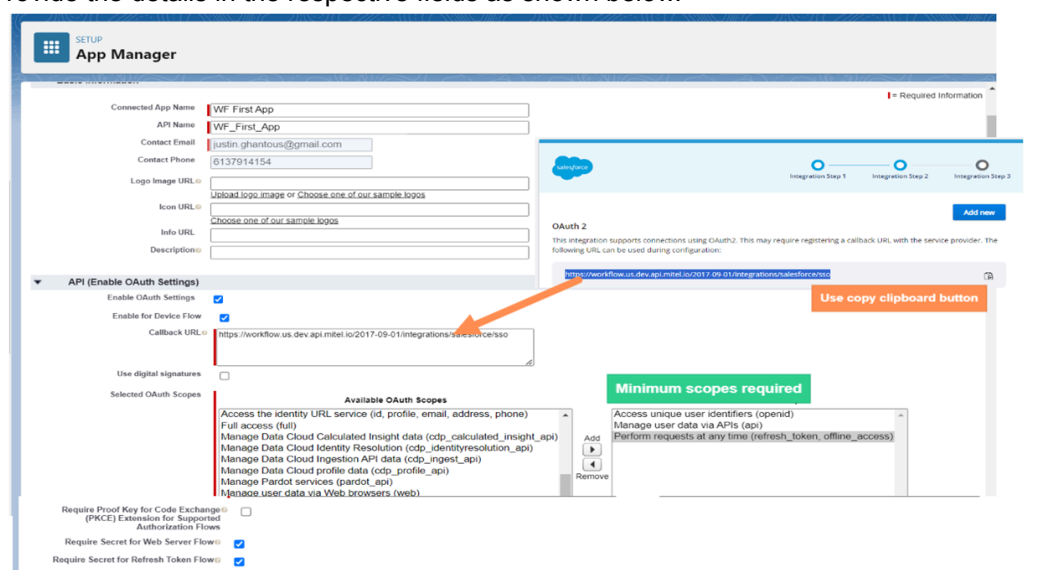


Figure 141: App Manager - New Connected App

5. Click **Save and Continue**.

6. Copy the values from the **Consumer Details** and save them securely for later use.

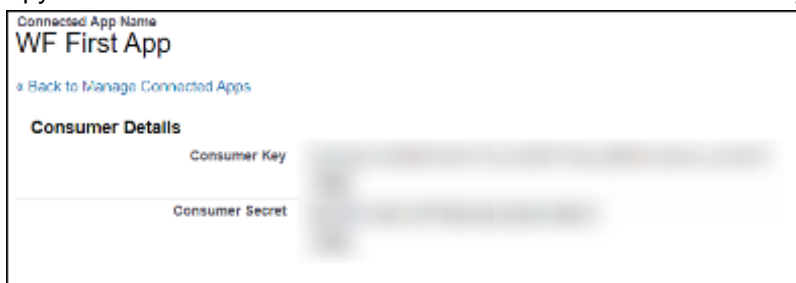

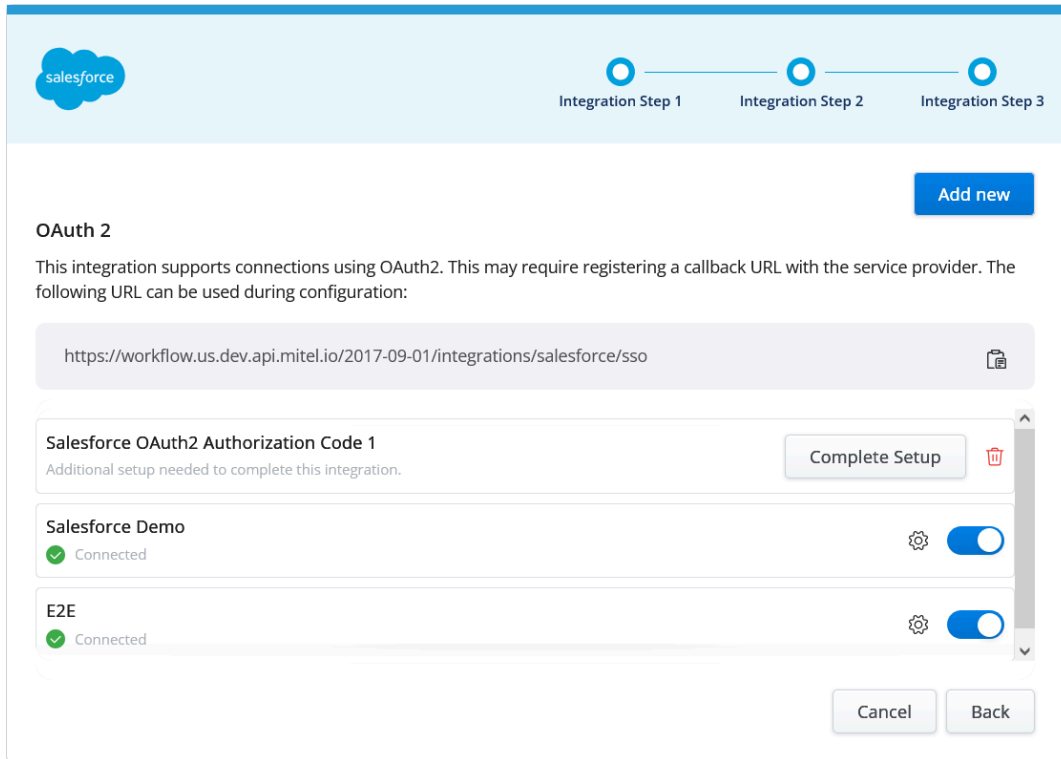


Figure 142: WF First App

9.13.2 Create Salesforce Integration

To add a Salesforce Workflow Integration, perform the following steps:

1. Login to **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click  icon on the **Salesforce** integration tile.
4. Click **Add new** to create a new connection.



salesforce

Integration Step 1 Integration Step 2 Integration Step 3

Add new

OAuth 2

This integration supports connections using OAuth2. This may require registering a callback URL with the service provider. The following URL can be used during configuration:

`https://workflow.us.dev.api.mitel.io/2017-09-01/integrations/salesforce/sso`

Complete Setup

Salesforce OAuth2 Authorization Code 1
Additional setup needed to complete this integration.

Salesforce Demo
✓ Connected

E2E
✓ Connected

Cancel **Back**

Figure 143: Salesforce Integration - Add new

5. Click **Complete Setup**.

Figure 144: Salesforce Integration - Configure this connection

6. In the **Configure this connection** screen, provide the following details:

- a. Enter an appropriate connection name in the **Name** field.
- b. Enter the consumer key created in the Salesforce Portal in the **Consumer Key** field.
- c. Enter the consumer secret created in the Salesforce Portal in the **Consumer secret** field.

7. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

8. Click **Continue**.

The newly created connection is displayed in the list. You can now use your Salesforce Integration in Workflow Studio.

9.13.3 Add a Salesforce activity

Once you have configured a connection, you can use the salesforce connection to perform various actions in a workflow. To configure a flow, perform the following steps:

1. In the **Workflow Studio**, open the new or existing flow to which you want to add this activity.

2. In the **Workflow Canvas**, open the activity wheel and search for “Salesforce” activities.

The following list of Salesforce actions is displayed.

- Salesforce Create or Update Record
- Salesforce Query SOQL
- Salesforce Get Record Field Values

Refer to the **Actions** section to view the definition of Salesforce actions.

3. Click on any one of the actions to add it to the flow.
4. Configure the parameters in the respective fields.
5. Click **Save** to save the flow.


9.14 Twilio Integration (Premier)

Twilio Integration requires the Account SID and Auth Token from your Twilio account.

Refer to the [Twilio](#) documentation for instructions on how to retrieve the information.

9.14.1 Create Twilio Integration

To create Twilio Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click  icon on the **Twilio** integration tile.

4. Click **Add new** to create a new connection.

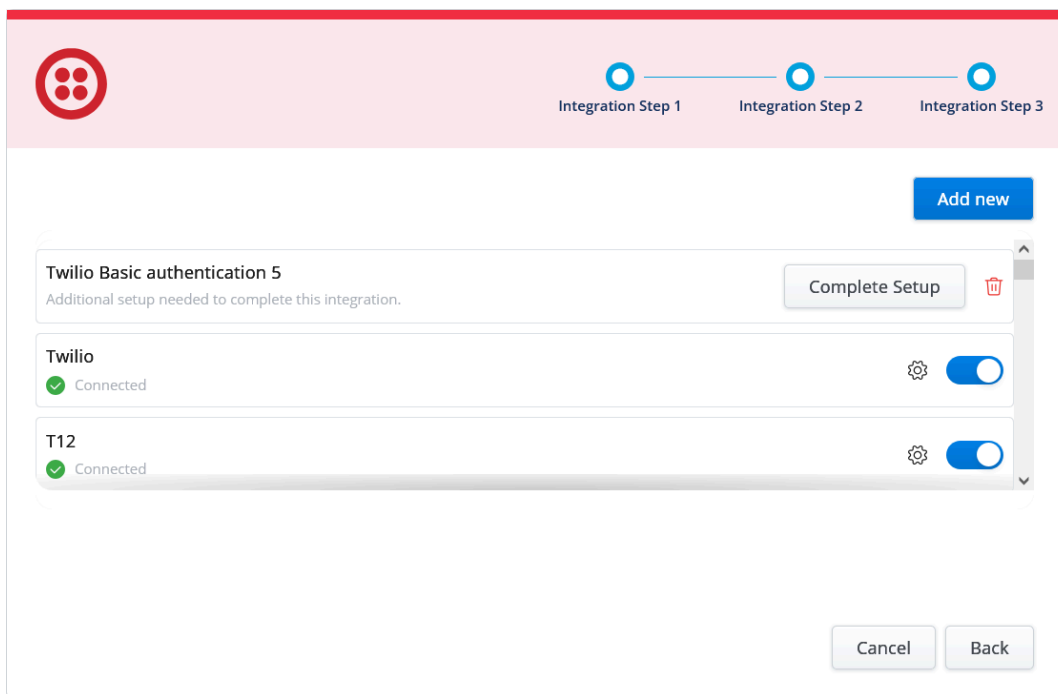


Figure 145: Twilio Integration - Add new

5. Click **Complete Setup** to provide a connection name and the product token provided from the CM.com website.

The screenshot shows a 'Configure this connection' dialog box. It has a title bar with a close button (X). The main content area contains three input fields: 'Name *' with the value 'Twilio Basic authentication 5', 'ACCOUNT SID *', and 'AUTH TOKEN *'. Below the 'ACCOUNT SID' field, the word 'Required' is written in red. Below the 'AUTH TOKEN' field, the word 'Required' is also written in red. At the bottom right, there is a blue 'Create' button.

Figure 146: Twilio Integration - Configure this connection

6. Click **Create**.


The **Connection Complete** screen is displayed to confirm the success of the third-party integration.


7. Click **Continue**.

The newly created connection is displayed in the list.

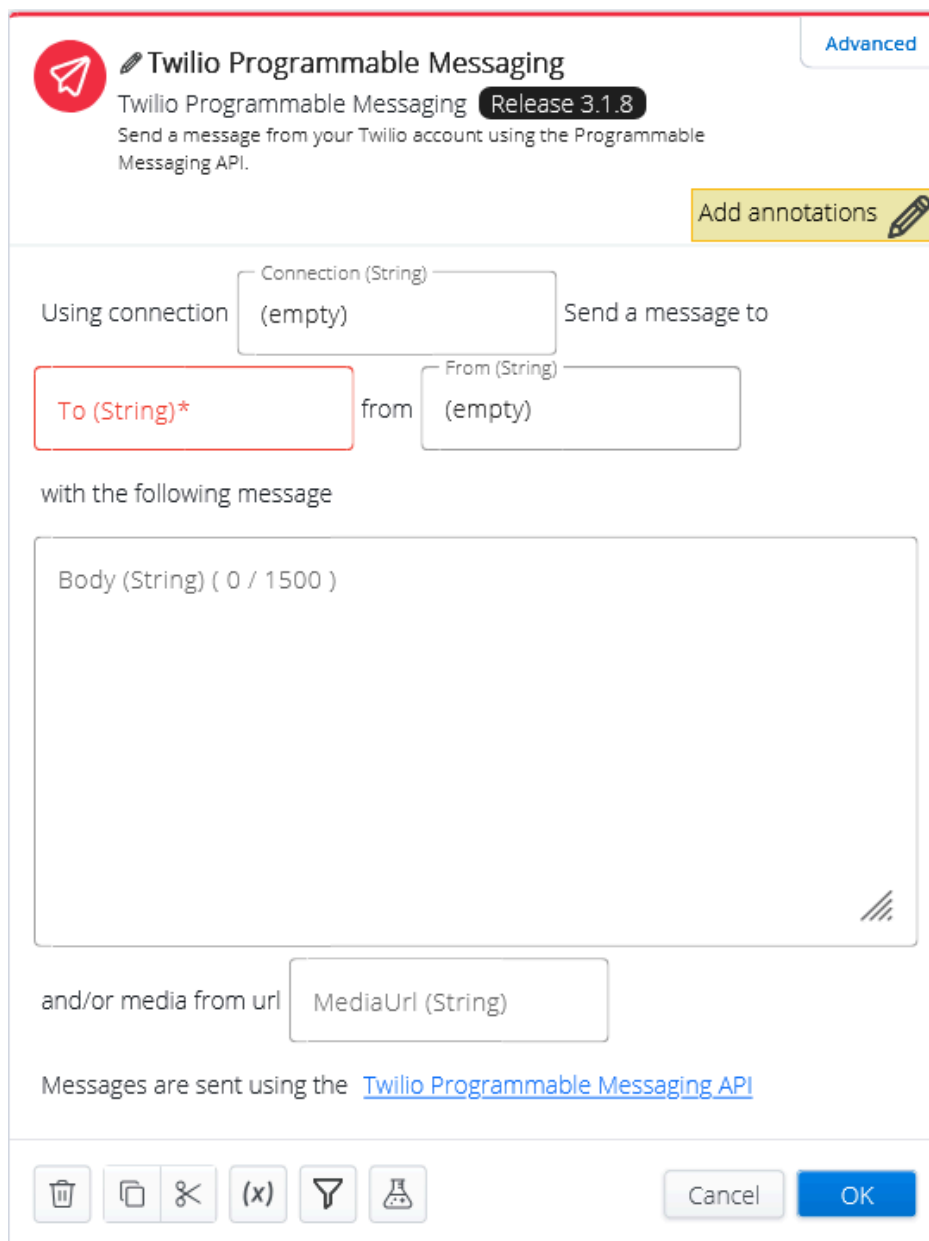
9.14.2 Add a Twilio send message activity

To add Twilio send message activity in a flow, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click  icon to add the new activity to the flow.

6. Under **Integration** activities, under  icon, click  to add the Twilio send message activity to the flow.

The **Twilio send message** popup screen is displayed.



Twilio Programmable Messaging Advanced

Twilio Programmable Messaging **Release 3.1.8**
Send a message from your Twilio account using the Programmable Messaging API.

[Add annotations](#)

Using connection Connection (String)
(empty) Send a message to

To (String)* from From (String)
(empty)

with the following message

Body (String) (0 / 1500)

and/or media from url MediaUrl (String)

Messages are sent using the [Twilio Programmable Messaging API](#)







      Cancel OK

Figure 147: Twilio send message

7. On the **Twilio send message** screen, provide the following details. For more information on fields, refer to the field description table.

**Note:**

Mandatory fields are marked with an asterisk (*).

Table 313: Twilio send message – Field Description

Field	Description
Connection	Select the connection created in the Twilio integration from the drop-down list.
To	Enter the destination phone number (E.164 format).
From	Select the registered mobile number from the drop-down list. The list is pre-populated based on the Twilio integration selected in the Connection drop-down list.
Message	Enter the message string\function that you would like to send to the recipient.
MediaURL	Enter the URL of the media that you want to send. It must be publicly accessible.

8. Click **OK** to add the Twilio send message activity to the flow.

9.15 Whispeak Integration (Premier)

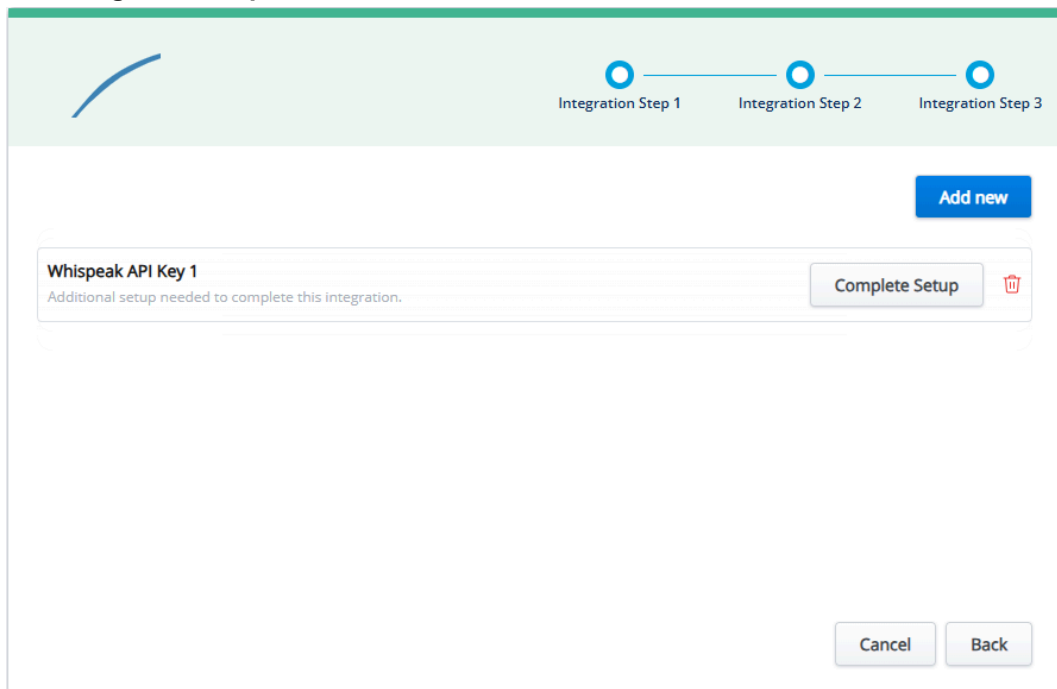
Whispeak Integration requires an API that can be generated from the Whispeak application. Refer to the [Whispeak](#) documentation for instructions on how to collect the required information.

9.15.1 Create Whispeak Integration

To create Whispeak Integration in Workflow Studio, perform the following steps:

1. Log into **Workflow Studio**.
2. Under **Library**, click **Integrations**.
3. In the **Integrations Hub**, click on the **See details** button on the **Whispeak** integration tile.
4. Click **Connect** to start the integration steps.
5. In the **Integration Step 1** screen, read through the Getting Started information and click **Next**.

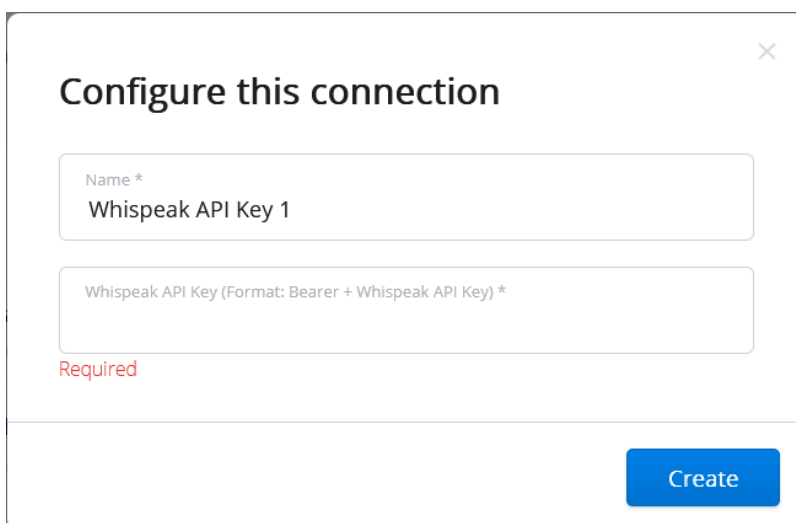
6. In the **Integration Step 2** screen, click **Add new** to create a new connection.



The screenshot shows the 'Integration Step 2' screen. At the top, there is a progress bar with three steps: 'Integration Step 1', 'Integration Step 2' (current), and 'Integration Step 3'. Below the progress bar, there is a blue 'Add new' button. Below the button, there is a list of existing connections. The first connection is 'Whispeak API Key 1' with the status 'Additional setup needed to complete this integration.' and a 'Complete Setup' button. At the bottom right, there are 'Cancel' and 'Back' buttons.

Figure 148: Whispeak Integration - Add new

7. Click **Complete Setup** to provide a connection name and the Whispeak API key.



The screenshot shows a 'Configure this connection' dialog box. It has a title bar with a close button. The main content area has two input fields: 'Name *' with the value 'Whispeak API Key 1' and 'Whispeak API Key (Format: Bearer + Whispeak API Key) *'. Below the second field, there is a red 'Required' label. At the bottom right, there is a blue 'Create' button.

Figure 149: Whispeak Integration - Configure this connection

8. Click **Create**.


The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

9. Click **Continue**.

The newly created connection is displayed in the list.

9.15.2 Add a Whispeak Create Signature activity

To add a Whispeak Create Signature activity in a flow, perform the following steps:

1. Navigate to the **Workflow Editor** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5. Click  icon to add the new activity to the flow.
6. In the **Search actions** bar, search and select the **Whispeak Create Signature** activity.
7. Click **Add**.

The **Whispeak Create Signature** pop up screen is displayed.

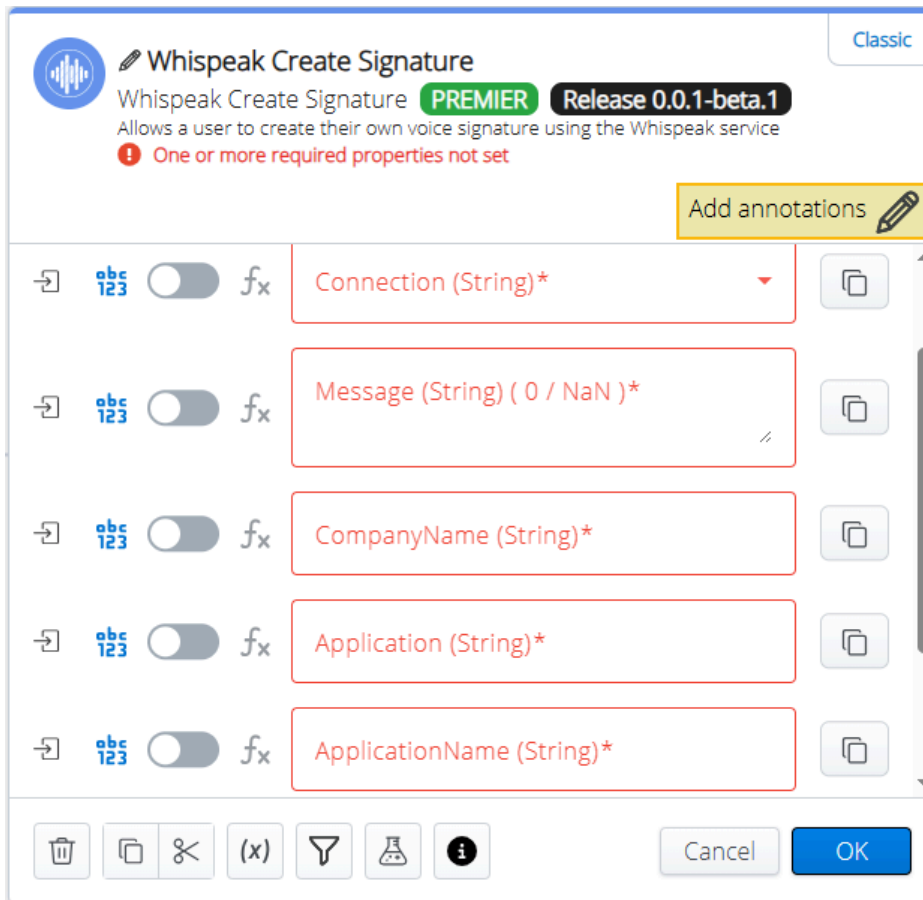


Figure 150: Whispeak Create Signature

8. On the **Whispeak Create Signature** screen, provide the following details. For more information on fields, refer to the field description table.

**Note:**

Mandatory fields are marked with an asterisk (*).



Table 314: Whispeak Create Signature – Field Description

Field	Description
Connection*	Select the connection created in the Whispeak integration from the drop-down list.
Message*	Enter the message that will be played to the caller to create their signature. For example, “Thank you for calling. You will create your voice signature after the beep. Continue to speak until you hear a second beep.”
Company Name*	Enter the company name provided by Whispeak.
Application*	Enter the application string provided by Whispeak.
Application Name*	Enter the application name provided by Whispeak.

9. Click **OK** to add the **Whispeak Create Signature** activity to the flow.

9.15.3 Add a Whispeak Authenticate activity

To add a Whispeak Authenticate activity in a flow, perform the following steps:

1. Navigate to the **Workflow Editor** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5.  Click  icon to add the new activity to the flow.
6. In the **Search actions** bar, search and select the **Whispeak Authenticate** activity.

7. Click **Add**.

The **Whispeak Authenticate** pop up screen is displayed.

Figure 151: Whispeak Authenticate

8. On the **Whispeak Authenticate** screen, provide the following details. For more information on fields, refer to the field description table.

Note:
Mandatory fields are marked with an asterisk (*).

Table 315: Whispeak Authenticate – Field Description



Field	Description
Connection*	Select the connection created in the Whispeak integration from the drop-down list.

Field	Description
Message*	Enter the message that will be played to the caller to authenticate their previously created voice signature. For example, "Thank you for calling. We have detected that you have created a voice signature. Please speak after the beep and continue to speak until you hear a second beep."
Signature ID*	The signature ID that was previously created
Company Name*	Enter the company name provided by Whispeak.
Application*	Enter the application string provided by Whispeak.
Application Name*	Enter the application name provided by Whispeak.

9. Click **OK** to add the **Whispeak Authenticate** activity to the flow.

9.15.4 Add a Whispeak Delete Signature activity

To add a Whispeak Delete Signature activity in a flow, perform the following steps:

1. Navigate to the **Workflow Editor** using **Create flow**.
2. Search and select a trigger in the **Search activities** search bar.
3. Click **Add**.
4. Update the required parameters and click **OK**.
5.  Click  icon to add the new activity to the flow.
6. In the **Search action** bar, search and select the **Whispeak Delete Signature** activity.

7. Click **Add**.

The **Whispeak Delete Signature** pop up screen is displayed.

Figure 152: Whispeak Delete Signature

8. On the **Whispeak Delete Signature** screen, provide the following details. For more information on fields, refer to the field description table.



Note:

Mandatory fields are marked with an asterisk (*).

Table 316: Whispeak Delete Signature – Field Description

Field	Description
Connection*	Select the connection created in the Whispeak integration from the drop-down list.
Signature ID*	Enter the signature ID that was previously created.

Field	Description
Company Name*	Enter the company name provided by Whispeak.
Application*	Enter the application string provided by Whispeak.
Application Name*	Enter the application name provided by Whispeak.

9. Click **OK** to add the Whispeak Delete Signature activity to the flow.

9.15.5 Sample Use Case

It is recommended to use a voice-based workflow with your Whispeak Integration. The following example illustrates how to configure a workflow to detect whether a user has created a voice signature. If the user does have a voice signature, the system will allow them to authenticate. If the user does not have a voice signature, the system will allow them to create a voice signature, which is used in subsequent calls.

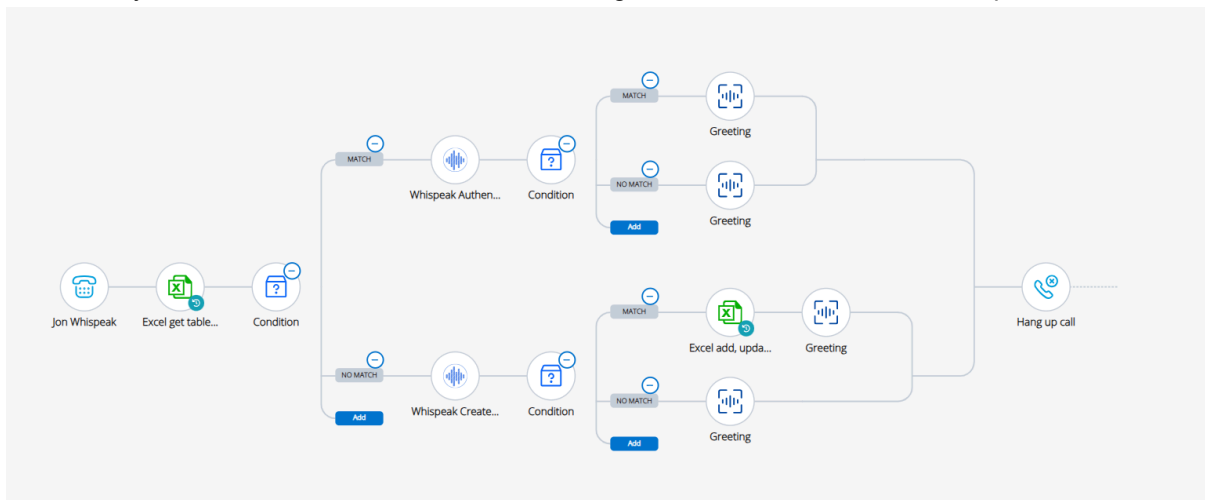


Figure 153: Sample Whispeak Workflow

To create the sample workflow for the above-mentioned use case, perform the following steps:

1. Use an **Endpoint Ringing** trigger.
2. A data source is required to store the user's phone number and Signature ID from Whispeak. In this example, the **Excel get table** activity is used.
3. Use a **Condition** activity to check whether the user has a voice Signature in the **Excel** table.
 - a. If the user has a voice Signature, use the **Whispeak Authenticate** activity.
The **Success** output will determine if the user was authenticated successfully.
 - b. If the user does not have a voice Signature, use the **Whispeak Create Signature** activity.
The **SignatureID** will determine if the user's voice signature was successfully created.
4. Create the user's row in Excel alongside their **Phone Number** and **SignatureID**.

5. Validate that the user's data is stored in **Excel** and optionally play a success or failure message.

9.16 Zoom Integration (Essential)

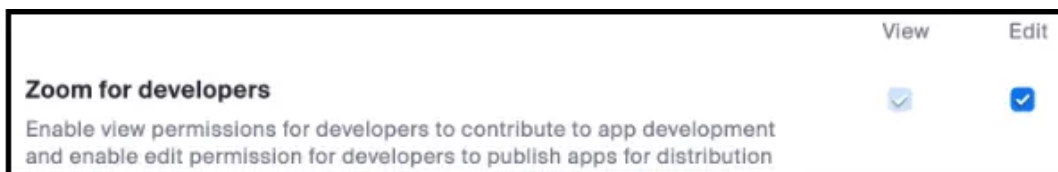
Zoom Integration is required to use the Zoom activities in **Workflow Studio**.

9.16.1 Prerequisites

To integrate Workflow Studio with the Zoom application, you must have a Zoom account assigned to you with the **Zoom for developers** role.

To enable the **Zoom for developers** role, perform the following steps:

1. Log into the Zoom web portal (<https://www.zoom.com>) as admin.
2. Under **User Management**, click **Roles**.
3. Select the role you want to edit or create a new role.
4. Under **Role Settings**, select **Advanced features**
5. Enable the **View** and **Edit** checkboxes for **Zoom for developers** role.



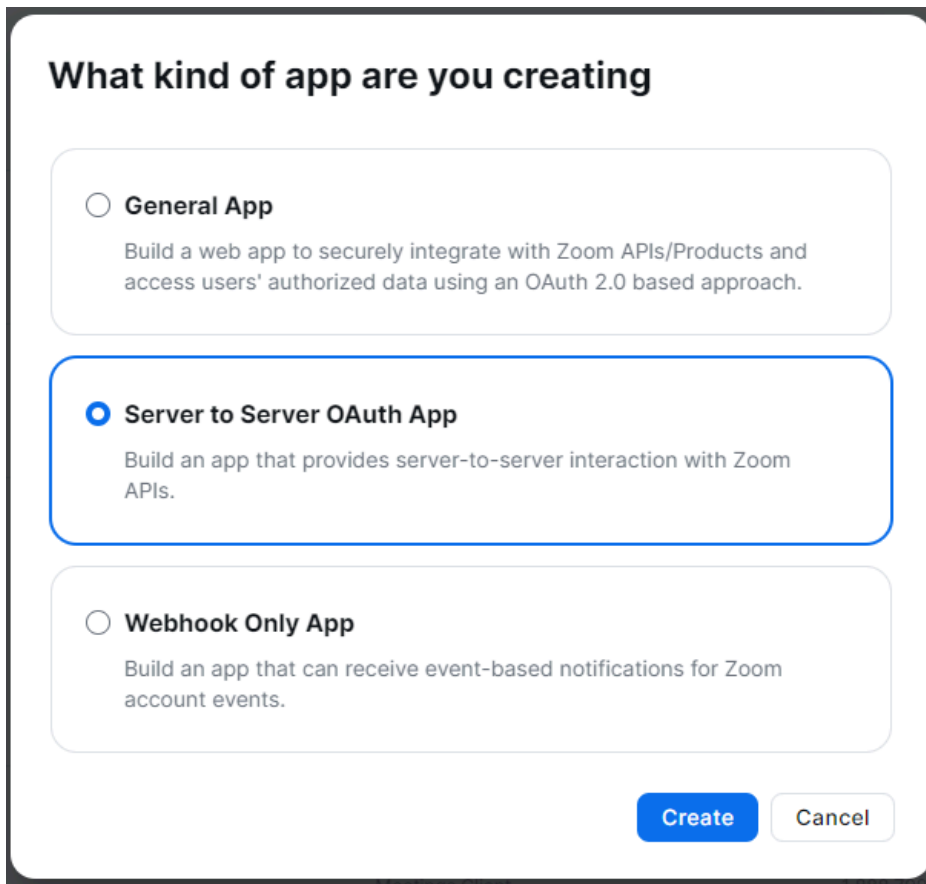
9.16.2 Create a New Application in Zoom

To create a new application in Zoom, perform the following steps:

1. Log in to **Zoom App Marketplace**.
2. Select **Develop** from the menu in the top right of the screen.

3. Click Build App.

The **Create Application** popup screen is displayed.



What kind of app are you creating

☐ **General App**
Build a web app to securely integrate with Zoom APIs/Products and access users' authorized data using an OAuth 2.0 based approach.

☒ **Server to Server OAuth App**
Build an app that provides server-to-server interaction with Zoom APIs.

☐ **Webhook Only App**
Build an app that can receive event-based notifications for Zoom account events.

Create **Cancel**

Figure 154: Create Application

4. Select Server to Server OAuth App from the option list.**5. Click Create.****6. Define an App Name for your application.**

7. Click **Continue**.

The **App Credentials** screen is displayed with the **Account ID**, **Client ID**, and **Client Secret** assigned to your application. This information will be required when integrating Zoom into Workflow.

App Credentials

Use the credentials to access Zoom APIs from your app. Make sure to securely store the credentials. Do not store them in public repositories.

App Credentials

Account ID

Copy

Client ID

Copy

Client Secret

Copy

Regenerate

Continue

Figure 155: App Credentials

8. Select **Continue**.

The **Information** screen is displayed.

Information

Basic Information

App Name

17/50

Short Description

Describe your app's core features in 1-2 sentences. This will show on the app listing card.

13/150

Company Name

Figure 156: Information

9. Enter any descriptive information for your application.

10. Click **Continue**.

The **Feature** screen is displayed with the secret token assigned to your application. This is required to add a Zoom trigger into your workflow.

11. Click **Continue**.

The **Add Scopes** screen is displayed.

Add Scopes

Scopes define the API methods this app is allowed to call, and thus which information and capabilities are available on Zoom. Scopes are restricted to specific resources like channels or files. If your app is submitted to Zoom, we will review your request for each scope. After your app is live, it will only be able to use permission scopes that Zoom approved. [Learn more about Zoom's scopes](#).

+ Add Scopes

Scopes	Description	Optional	
User			
View all user information			
user:read:list_collaboration_devices:admin	View a user's collaboration devices	<input type="checkbox"/>	
user:read:collaboration_device:admin	View a user's collaboration device	<input type="checkbox"/>	
user:read:email:admin	Verify a user's email	<input type="checkbox"/>	
user:read:list_assistants:admin	View a user's assistants	<input type="checkbox"/>	
user:read:clocked_in_user:admin	Clock in a user	<input type="checkbox"/>	

Figure 157: Add Scopes

The scopes determine which Zoom API calls are allowed when using your application. For details on scopes and their access, refer to the [Zoom Developer](#) documentation.

12. Click **Continue**.

The **Activation** screen is displayed to activate the application.

The new application has been created and is available in **Workflow Studio**.

9.16.3 Create Zoom Integration

To create Zoom integration into **Workflow Studio**, perform the following steps:

- 1. Log into **Workflow Studio**.
- 2. Under **Library**, click **Integrations**.

The **Integrations Hub** screen is displayed.

- 3. In the **Integrations Hub**, click **See details** next to the **Zoom** integration tile.
- 4. Click **Connect**.

5. Click **Next**.

The **Zoom Integration - Add new** screen is displayed.

Integration Step 1 Integration Step 2 Integration Step 3

Add new

OAuth 2

This integration supports connections using OAuth2. This may require registering a callback URL with the service provider. The following URL can be used during configuration:

<https://workflow.us.dev.api.mitel.io/2017-09-01/integrations/zoom/sso>

Zoom OAuth2 Client - Justin's account

✓ Connected

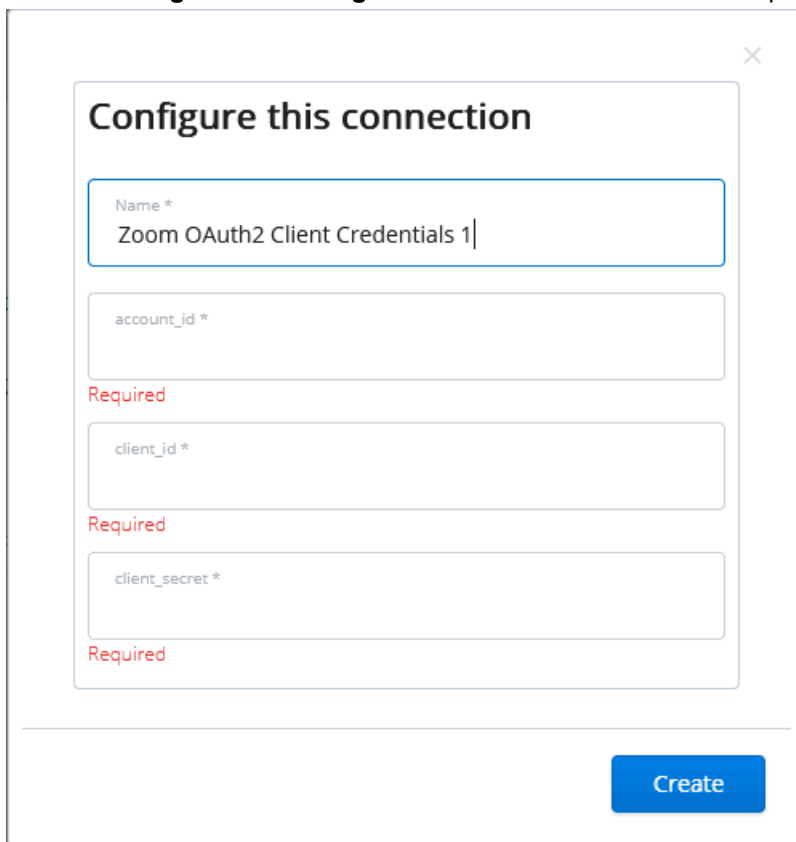
Cancel Back

Figure 158: Zoom Integration - Add new

6. Click **Add new** to add a new connection.

7. Click **Complete Setup** to create a new connection.

The **Zoom Integration - Configure this connection** screen is displayed.



Configure this connection

Name *
Zoom OAuth2 Client Credentials 1

account_id *
Required

client_id *
Required

client_secret *
Required

Create

Figure 159: Zoom Integration - Configure this connection

8. In the **Configure this connection** screen, enter the following details.
 - a. In the **Name** field, enter a descriptive name for the connection.
 - b. In the respective fields, enter the **account_id**, **client_id**, and **client_secret** from the Zoom **App Credentials** page.
9. Click **Create**.

The **Connection Complete** screen is displayed to confirm the success of the third-party integration.

10. Click **Continue**.

The newly created connection is displayed in the list.



Zoom OAuth2 Client - Justin's account

✓ Connected

⚙️ ☒

If not, click on the gear icon and confirm that the **account_id**, **client_id**, and **client_secret** are entered correctly.

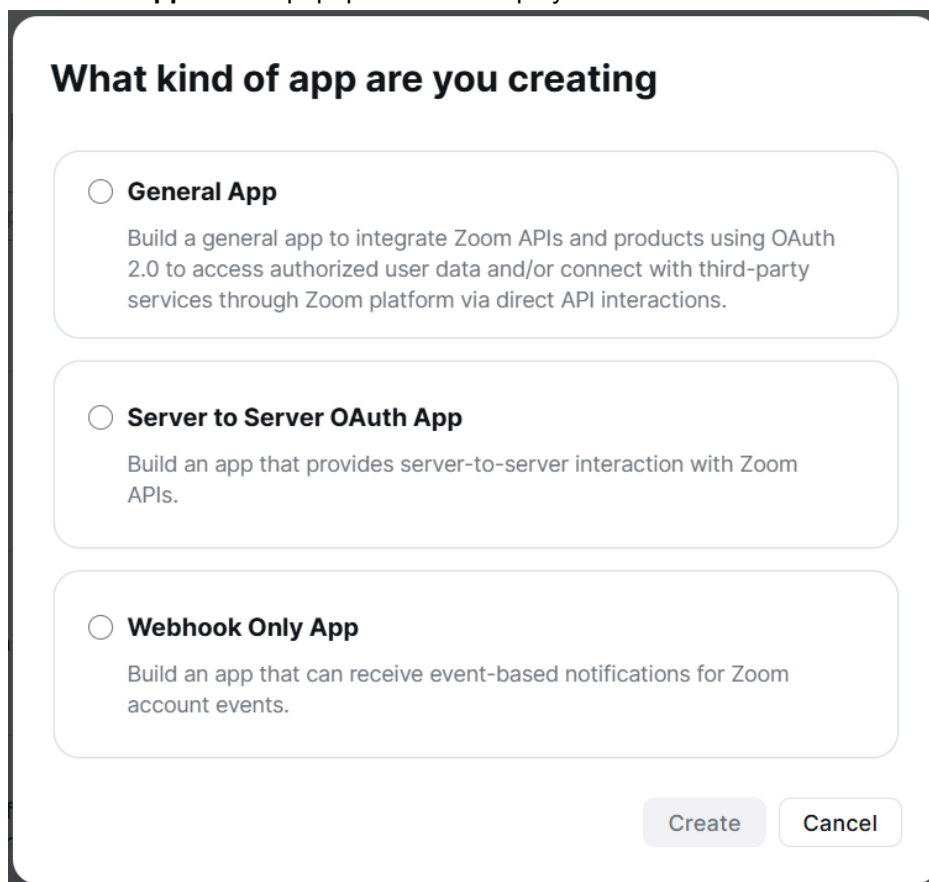
9.16.4 Configure the Trigger Notification in the Zoom App

Zoom trigger allows you to receive notifications when an event is generated within the Zoom ecosystem. Many events within Zoom can be configured to notify the Zoom trigger that a change has occurred.

To configure the trigger notification event in the Zoom application, perform the following steps:

1. Log in to the **Zoom App Marketplace** as an administrator.
2. Select **Develop** from the menu in the top right of the screen.
3. Click **Build an App**.

The **Create Application** popup screen is displayed.



The screenshot shows a modal dialog titled "What kind of app are you creating". It contains three radio button options, each with a description:

- ☐ **General App**
Build a general app to integrate Zoom APIs and products using OAuth 2.0 to access authorized user data and/or connect with third-party services through Zoom platform via direct API interactions.
- ☐ **Server to Server OAuth App**
Build an app that provides server-to-server interaction with Zoom APIs.
- ☐ **Webhook Only App**
Build an app that can receive event-based notifications for Zoom account events.

At the bottom right of the dialog are two buttons: "Create" and "Cancel".

Figure 160: Create Application

4. Select **Webhook Only App** from the option list.

5. Click **Create**.

The **Create a Webhook app** screen is displayed.

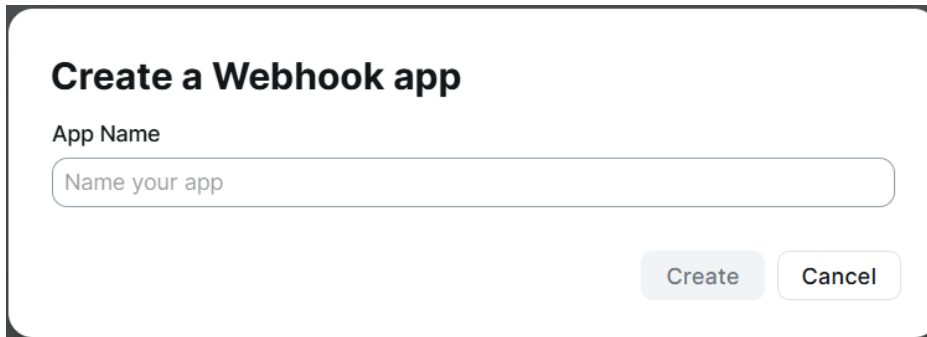


Figure 161: Create a Webhook app

6. Provide an **App Name** for your application.7. Click **Create**.

The **Information** tab is displayed.

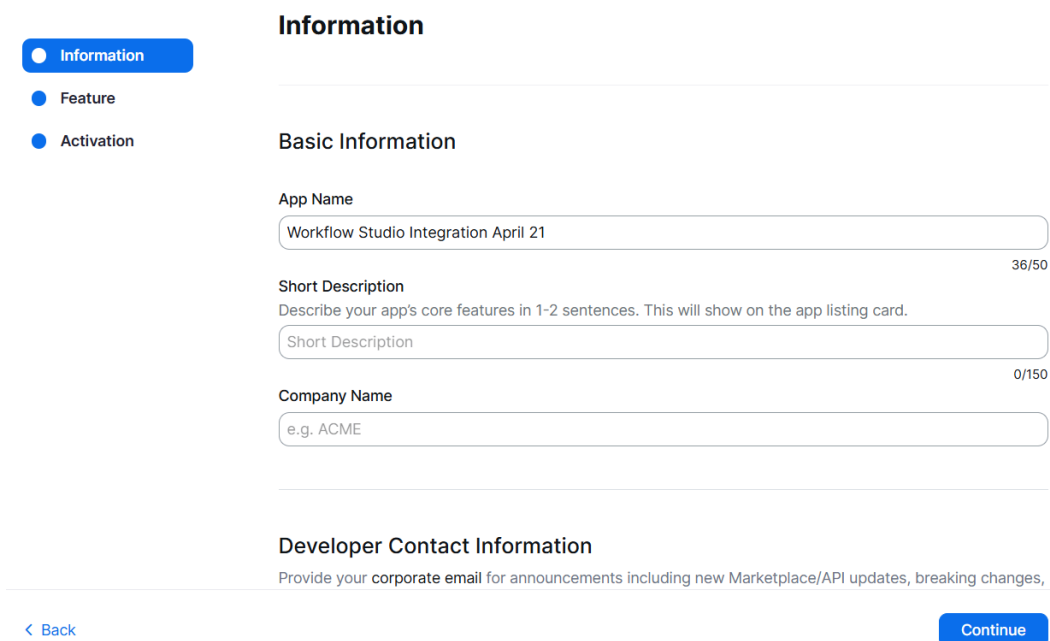


Figure 162: Information

8. In the **Information** tab, enter the details for your application.9. Click **Continue**.

The **Features** tab is displayed with the secret token assigned to your application.

10. Copy the Secret Token to your clipboard. You will use it when setting up the Zoom Trigger flow within the Workflow Studio.

11. Enable **Subscription**.

The **Subscription** screen is displayed.

Subscription name (optional)
Name this particular event subscription

WF April 21 Sub

Event notification endpoint URL
Provide a URL to receive subscribed event notifications. This is for your production environment.

https://workflow.us.dev.api.mitel.io/2017-09-01/webhooks/accounts/66d43e89-b3d2-477e-l

Validate the URL
✔ Validated

Authentication Header Option (Optional) >
Use the custom authentication header method to validate Zoom webhook notifications.

Add Events
Add events for your app to subscribe.

+ 1 events added

Save Cancel

Figure 163: Subscription

12. Provide a subscription name.

13. In the **Event Notification endpoint URL** field, enter the Workflow URL Webhook.

14. Click the **Validate** button.

If the validation is successful, the **Validate** button turns green as shown above.

If the validation fails, the trigger is not configured to execute.

Note:

Ensure that the URL copied from Workflow Studio is complete and includes `workers?sync=true`.

15. After validation is successful, you will be able to add Events that you want to listen to. In the above example, the event for listening to presence changes of a user is shown.

16. Click **Add Events**.

The **Event Types** screen is displayed.

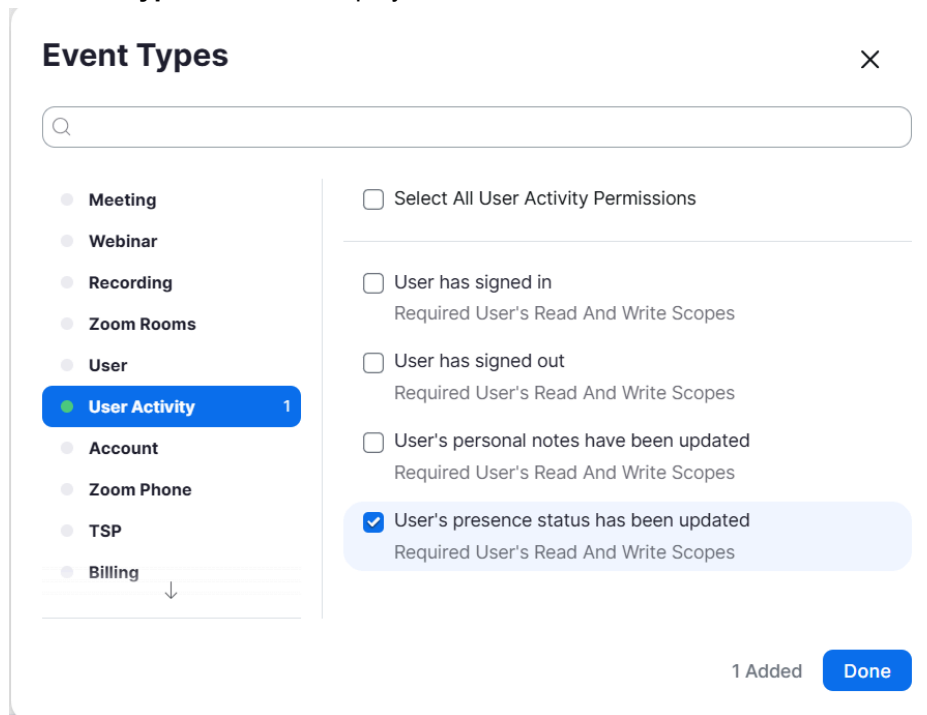


Figure 164: Event Types

17. Under the **User Activity** tab, select the **User's presence status has been updated** checkbox.

18. Click **Done**.

The **User Presence Event** gets added.

19. Click **Save**.

You can now create a workflow to perform actions whenever a user's presence changes within Zoom.

To create a workflow that triggers with a Zoom change, perform the following steps:

1. Create a workflow and add the **Zoom Trigger**.

2. Click on the trigger.

The **Zoom Trigger** screen is displayed.

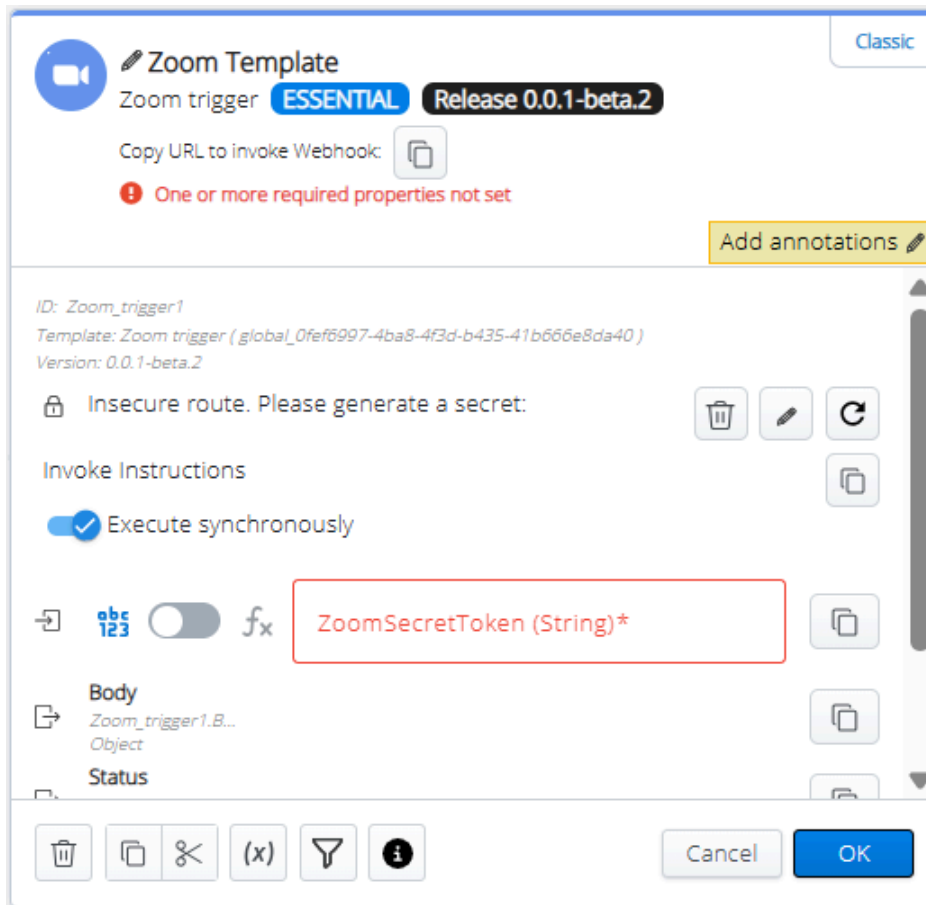


Figure 165: Zoom Trigger

3. Enable **Execute Synchronously**.
4. Enter the Zoom Secret Token provided when you set up the Zoom App within the Zoom Portal.
5. Click **OK**.

9.16.5 Add a Zoom Trigger

To add a Zoom trigger to a flow, perform the following steps:

1. Navigate to the **Workflow Canvas** using **Create flow**.

The **Workflow Canvas** is displayed with the **Search triggers** form.

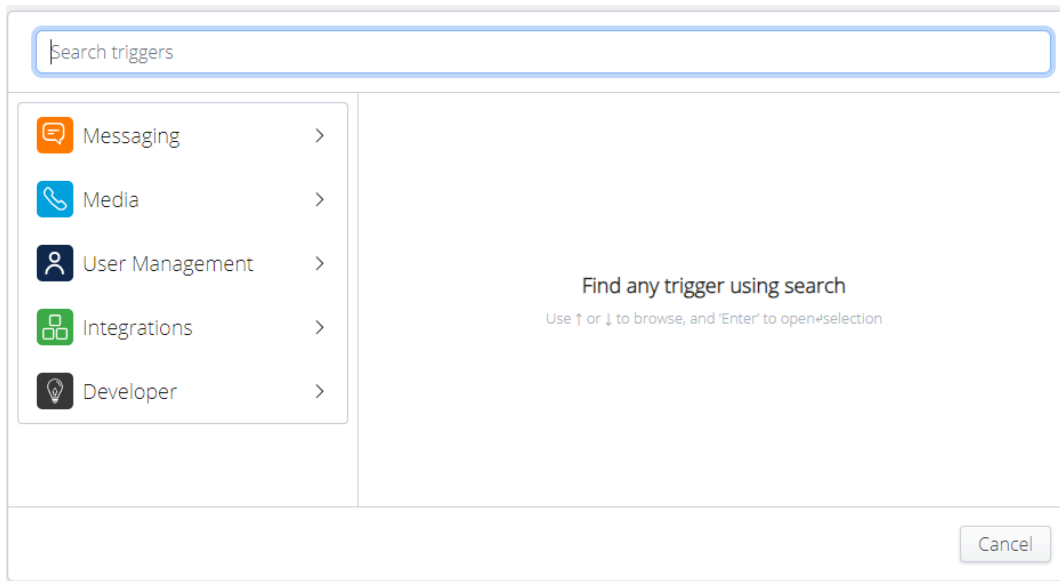


Figure 166: Workflow Canvas - Search triggers

2. Search and select a **Zoom Trigger** under the **Integrations** section.

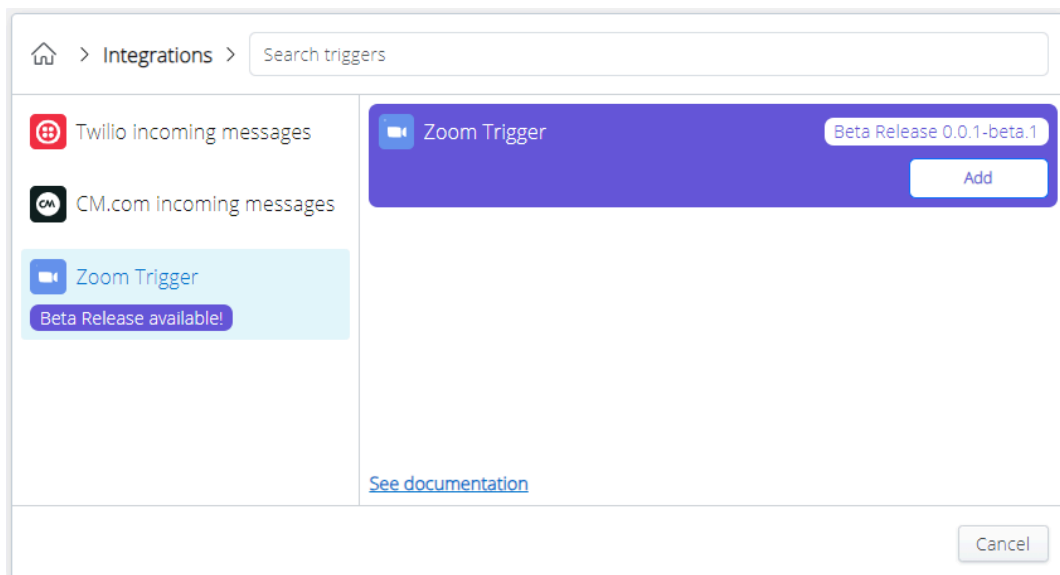


Figure 167: Add Zoom Trigger

3. Click **Add** to add the trigger to the workflow.

The **Zoom Trigger** popup screen is displayed.

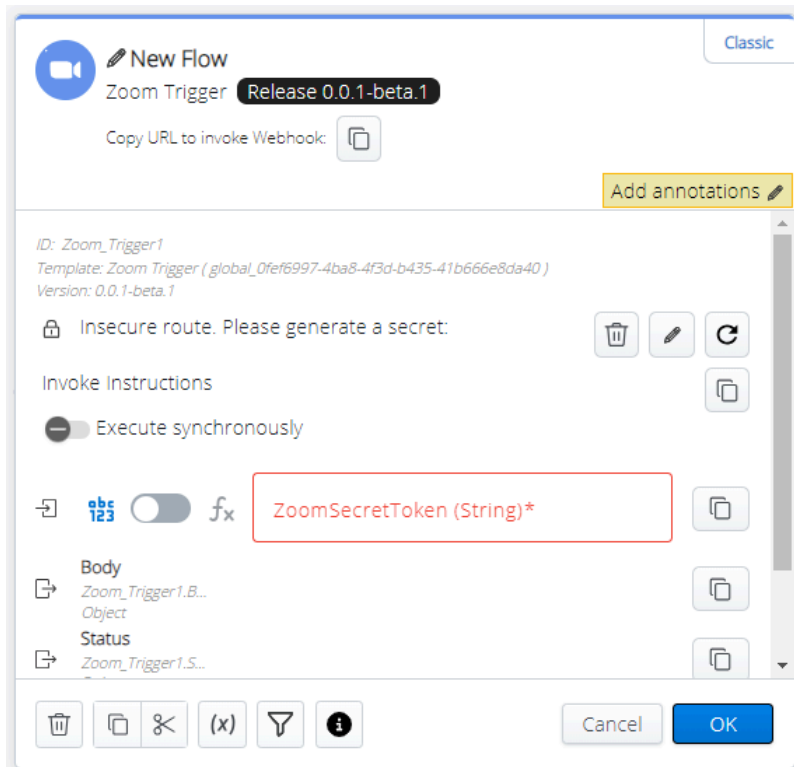


Figure 168: Zoom Trigger

4. In the **ZoomSecretToken** field, enter the secret token generated in the **Zoom App Feature** page.

For more details on other options for the trigger, refer to the **Webhook** description.

5. Click **OK**.

The **Zoom Trigger** is added to the flow.

9.16.6 Use the Zoom API in Workflow

To make REST API calls into Zoom from a workflow, perform the following steps:

1. In the **Workflow Studio**, open the new or existing flow to which you want to add the REST API call activity.

2. In the **Workflow Canvas**, open the activity wheel.



Figure 169: Search Activity Wheel

3. Under the **Developer** category, click on the **HTTP Request** activity.

The **HTTP Request** activity screen is displayed.

Http request Release 1.0.0
Integrate with RESTful Web APIs

Advanced

Add annotations

GET Test

Headers Body Authorization Testing

Integration

Connection

Cancel OK

Figure 170: HTTP Request

4. In the **Http request** activity, click on the **Authorization** tab.
5. In the **Integration** field, select the **Zoom** option from the list.
6. In the **Connection** field, select the connection you created from the list.

7. Enter the Zoom API you will be using.

In the above example, **GET <https://api.zoom.us/v2/users>** is used to retrieve all users in the account.

8. Click on the **Testing** tab to confirm that the connection is set up correctly to call the API.

The results appear in the **Response** area of the tab.

If there is an error, check the details of the error returned and verify that the application added in Zoom has the appropriate scope value to access the called API.

This chapter contains the following sections:

- [How to configure the Dial Menu timeout settings and destination](#)
- [How to configure the Dial Menu transfer to a voicemail box](#)
- [How to configure the Replay Dial Menu Greeting](#)
- [How to configure the hang up on caller activity](#)
- [How to configure the business opening hours for public holidays](#)
- [How to reuse activities in many flows](#)

10.1 How to configure the Dial Menu timeout settings and destination

To configure a timeout setting via the Dial Menu Activity, perform the following steps:

1. Click on the **Advanced** tab.

2. In the **EnterDigitTimeoutSeconds** field, modify the value to determine the time between digits that are selected.

If the minimum and maximum digits are set to 1, it will wait for the **EnterDigitTimeoutSeconds** duration before going down the timeout branch.

Figure 171: Dial Menu Timeout Settings

10.2 How to configure the Dial Menu transfer to a voicemail box

To transfer the call based on the digits entered in the dial menu, which can be a mailbox, extension, or any dialable number on the PBX, perform the following steps:

1. Add a **Dial Menu** Activity.
2. Set your minimum and maximum digits.
3. Remove the **MATCH** branch and have only a **NO MATCH** and **FAILED** branch, as you will not be validating the digits submitted by the customer.
4. In the **NO MATCH** branch, click on the **Advanced** tab, and click the toggle switch next to the destination input field to toggle the option to an expression (fx).
5. Enter the activityName.CollectectedDigits, in this scenario, the **Dial Menu** activity name is Dial_menu1.

6. In the **FAILED** branch, add a greeting activity to provide customer feedback that something went wrong and to try again.

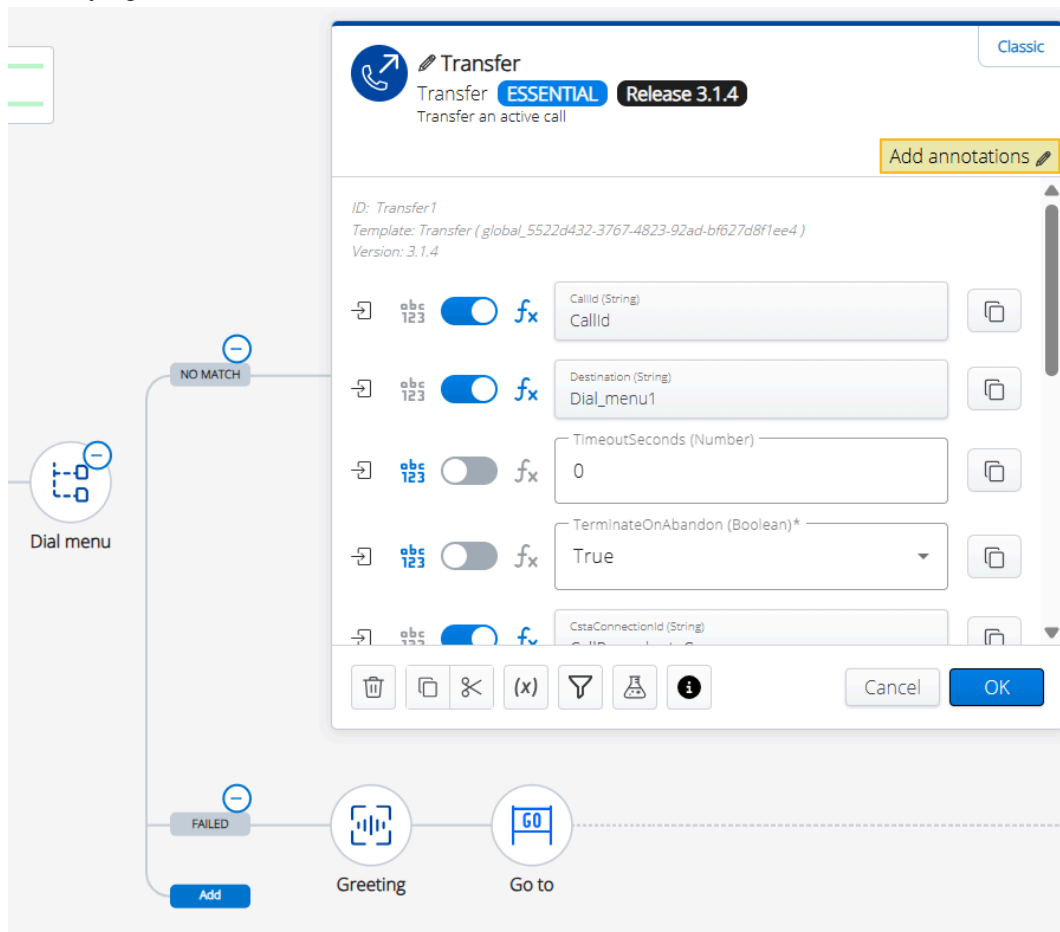


Figure 172: Dial Menu - Transfer Activity

10.3 How to configure the Replay Dial Menu Greeting

If no digits are selected within the specified time for greeting, the **Replay Dial Menu Greeting** can be configured.

To replay the dial menu, add a **Go To** activity to return to the **Dial Menu** in the Timeout Branch.

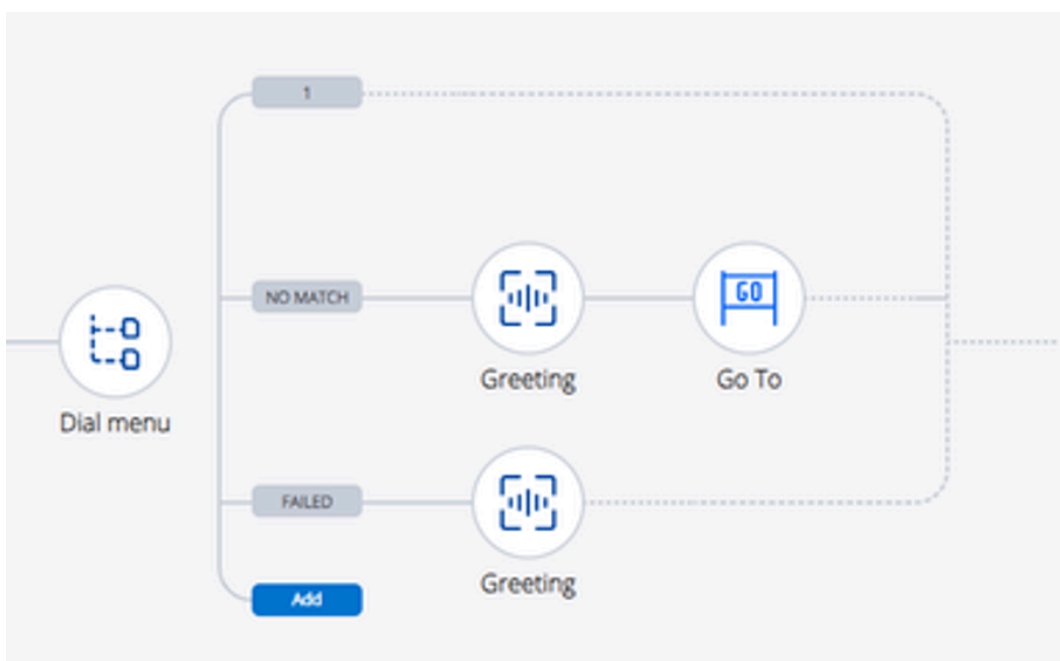


Figure 173: Replay Dial Menu Greeting

10.4 How to configure the hang up on caller activity

To configure hang up on caller if X number of dial menu replays are exceeded, perform the following steps:

1. In the **Dial Menu** activity, click the **Advanced** tab.
2. In the **RepeatTimes** field, enter the number of attempts that the system will allow the customer to hear before hanging up the call.

Once this number exceeds, it will follow the **FAILED** branch.

10.5 How to configure the business opening hours for public holidays

The following screenshot illustrates how to configure the business opening hours for public holidays.

1. In the **Schedule** activity, select the **Custom** option from the **Recurrence** drop-down list.

The screenshot shows the 'Schedule' activity configuration window. At the top, there is a title bar with a calendar icon, the text 'Schedule', and a version indicator 'Release 2.2.4'. A tab labeled 'Advanced' is visible in the top right corner. Below the title bar, there is a yellow button labeled 'Add annotations' with a pencil icon. The main configuration area is divided into two tabs: 'Schedule' (selected) and 'Exceptions from schedule'. Under the 'Schedule' tab, there is a 'Timezone' dropdown menu set to 'UTC -05:00 EST (Toronto)'. Below this, there is a 'Time Range' section with a 'Time of Day' dropdown set to 'Time of Day', and two input fields for 'From' (09:00) and 'To' (17:00). A minus button is located to the right of the 'To' field. Below the 'Time Range' section, there is a 'Recurrence' dropdown menu set to 'Custom...every week in May on Monday'. Below the 'Recurrence' dropdown, there is an 'Add..' button. At the bottom of the configuration area, there is a date and time field showing '2/18/2025, 12:41:20 PM' with a calendar icon, and a 'Test' button. The bottom of the window features a toolbar with icons for delete, copy, paste, undo, redo, and a help icon, followed by 'Cancel' and 'OK' buttons.

Figure 174: Schedule Activity

2. Select the day and month of your desired public holiday.

[Back](#)

Please enter a recurrence rule ...

every week in May on Monday

Repeat every

Interval

Frequency

Repeat on

Weekdays

Months

Set Position (nth occurrence with)

Figure 175: Recurrence Rule

3. In the **Set Position** field, enter the value as -1, which configures the date to always be the last selected day (Monday) of the selected month (May).
4. Click **OK**.

10.6 How to reuse activities in many flows

There are occurrences in which you create an activity in a flow that you would like to use in other flows. For example, adding a **Schedule** Activity inside a flow with many rules, like public holidays. To avoid copying this activity to many different flows and trying to manage all the flows if a schedule needs to be changed.

The recommended approach is to create an action with the **Schedule** activity and add it to multiple flows. If you modify the **Schedule** action, all flows referencing this action will execute using the updated changes.

This chapter contains the following sections:

- [Securing Data - Configuring and Executing Flows](#)
- [Configuration of Third-party Integrations](#)
- [Rate Limits](#)
- [Data Retention Schedule](#)
- [Compliance](#)

Security is a critical part of any application. This chapter provides information on how Workflow Studio ensures security in all aspects, from configuring third-party integration to collecting information from a caller.

11.1 Securing Data - Configuring and Executing Flows

Data that resides within a flow

Workflow Studio provides the ability to redact any information that is configured within a flow.

To redact the information, enable the **IsSecure** toggle while creating a new variable.



Figure 176: New Variable

This will ensure that the data in the UI is encrypted and not displayed as shown below:

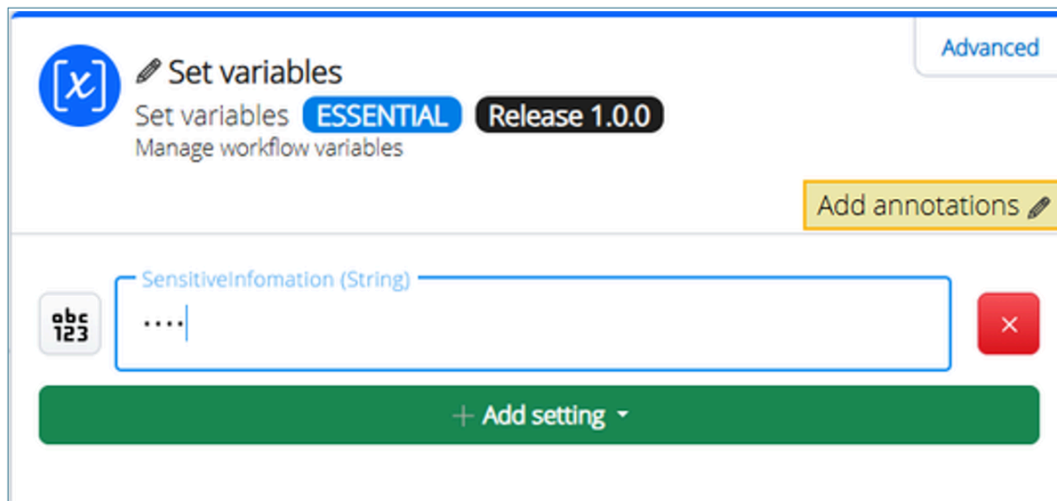


Figure 177: Set Variables

Data collection during flow execution

When a flow is executed, variables set to **IsSecure**, are encrypted when viewing the flow in Debug and the server logs. As shown below, the variable is encrypted when viewing the execution of the activity

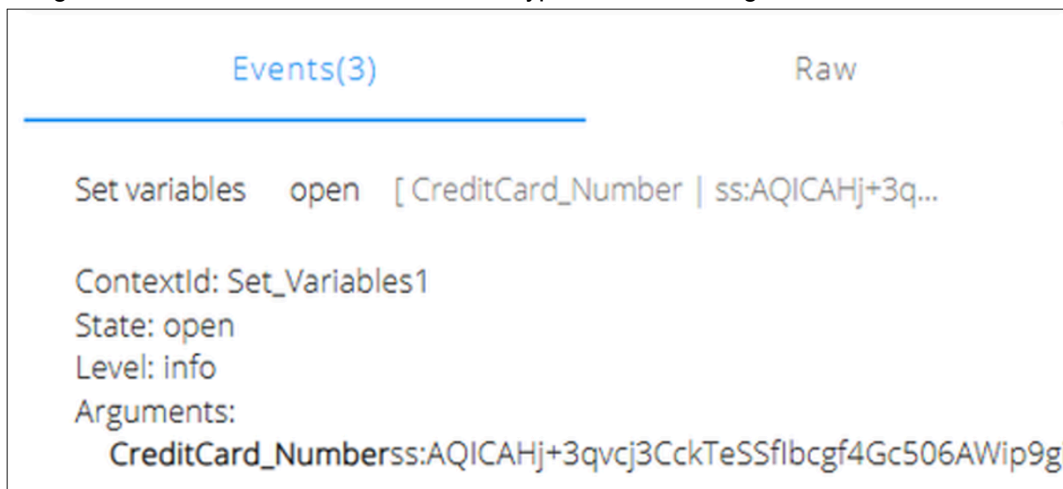


Figure 178: Encrypted Variables

Webhook Security-supported Methods

Webhooks are a very common way to communicate with third-party applications. However, there is a downside as the URL is globally available, malicious users can attempt to harm the system. It is recommended that all webhooks use one of the supported methods to secure the API call. Workflow Studios supports the following types of authentication.

To enable secure webhooks, click on the **Generate** button.

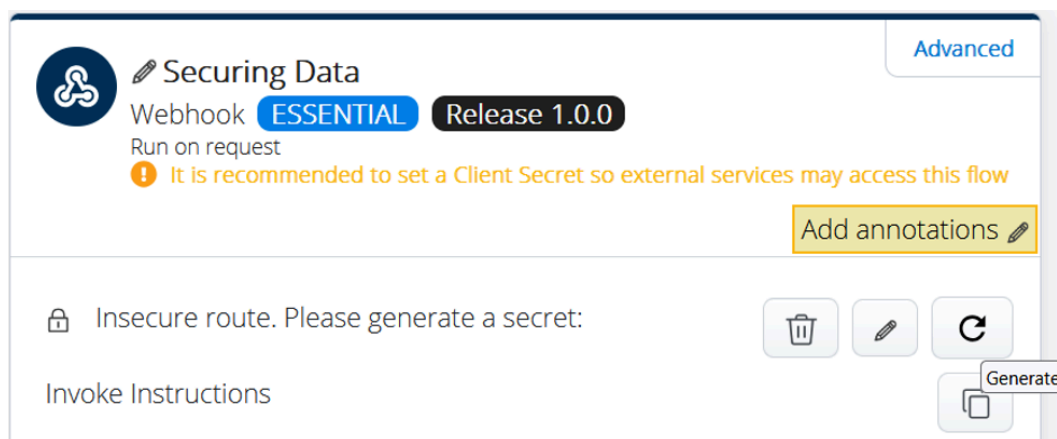


Figure 179: Insecure Route

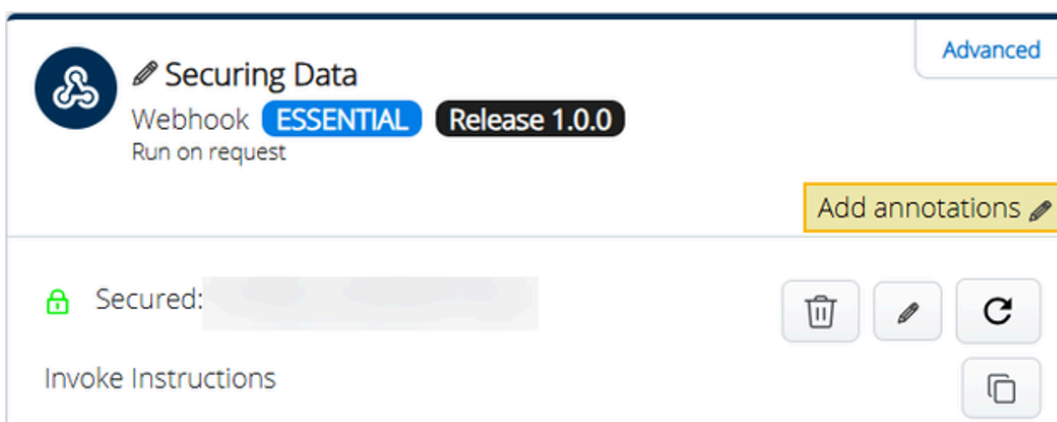


Figure 180: Secured Data

```
POST https://workflow.us.dev.api.mitel.io/2017-09-01/webhooks/accounts/
/activities/
/workers
Content-Type: application/json
x-mitel-shared-secret: 
```

Figure 181: Client Secrets - Example

```
70a247a0ce01/activities/
/workers
Content-Type: application/json
Authorization: Basic b64encoded... credentials as x-mitel-shared-secret:
{}
```

Figure 182: Basic Authorization - Example

```

POST https://workflow.us.dev.api.mitel.io/2017-09-
01/webhooks/accounts/[REDACTED]
[REDACTED]/activities/[REDACTED]
[REDACTED]2/workers
Content-Type: application/json
Authorization: Bearer [REDACTED]

```

Figure 183: Bearer Token – Example

11.2 Configuration of Third-party Integrations

Configuring integration connections for third-party applications requires sensitive data to be configured against the connection. This data could be API keys, client secrets, tokens, etc. Workflow takes the following steps to secure this information.

11.2.1 Storing of Client secrets/tokens/API Keys

Mitel Workflow Studio is designed with a strong emphasis on security to protect sensitive data provided by users when configuring Integration Connections. The following security measures ensure the confidentiality, integrity, and availability of sensitive credentials such as client secrets, tokens, and API keys:

Encryption and Secure Storage

- All sensitive data provided for Integration Connections is encrypted using **AWS Key Management Service (AWS KMS)** before being stored.
- AWS KMS provides highly secure, scalable, and fully managed encryption keys, ensuring robust protection against unauthorized access.
- Once encrypted, the sensitive data is securely stored within **AWS DynamoDB (DDB)**, a managed NoSQL database service with built-in security and compliance features.

Restricted API Access

- None of the sensitive data (client secrets, tokens, API keys) is ever exposed through Workflow APIs.
- Only authorized internal processes can access and use this encrypted information as required for integrations, ensuring zero direct exposure in workflows or user-accessible interfaces.

Access Controls and Compliance

- Access to encrypted credentials is tightly controlled using AWS Identity and Access Management (IAM) policies.
- Only essential services and authorized personnel can access the encrypted data, following the principle of least privilege.
- Mitel Workflow Studio follows industry best practices and compliance standards to ensure data security and regulatory adherence.

By implementing these security controls, **Mitel Workflow Studio** ensures that user credentials remain protected throughout their life cycle, providing a secure and reliable integration experience.

11.3 Rate Limits

Rate Limits Implementation in Workflow APIs

Rate limits are enforced on workflow APIs to maintain the security, stability, and performance of the **Mitel Workflow Studio** platform. These rate limits help prevent abuse, excessive requests, and potential security threats while ensuring fair usage across all users and integrations.

Implementation of Rate Limits

- Workflow APIs implement request throttling using AWS API Gateway Rate Limiting mechanisms.
- Limits are applied based on requests per second (RPS), ensuring APIs can handle a controlled volume of traffic.
- Requests exceeding the allowed rate receive HTTP 429 (Too Many Requests) responses, instructing clients to back off and retry after a cool down period.

Security Benefits of Rate Limiting

1. Protection Against DDoS Attacks

- Rate limits prevent Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) attacks by limiting the number of requests an entity can send within a specific time frame.
- This ensures that malicious users cannot overload the system and degrade service availability.

2. Prevention of Brute-Force Attacks

- API endpoints handling authentication, API key validation, or sensitive operations are protected against brute-force attempts by limiting login and authentication requests per user/IP.
- This helps prevent attackers from systematically trying multiple credentials to gain unauthorized access.

3. Ensuring Fair Resource Allocation

- Rate limiting prevents resource monopolization by a single user or integration, ensuring all users get fair access to API services.
- This helps maintain API performance and reliability across multiple concurrent users.

4. Mitigating API Misuse and Abuse

- Accidental or intentional overuse of API calls can degrade system performance; rate limits ensure that API consumers use resources efficiently.
- Prevents unintended infinite loops in automation scripts or integrations that might spam the API with unnecessary requests.

Implementation Details

- Rate limits are customizable per API endpoint based on usage patterns and security risks.
- Users exceeding the allowed limits can use exponential backoff and retry mechanisms to gracefully handle API call restrictions.

- API usage metrics and logs are monitored in real-time to detect and respond to anomalies or abuse attempts.

By enforcing rate limits, **Mitel Workflow Studio** enhances the security, performance, and availability of its Workflow APIs. These measures protect against malicious attacks, unintentional misuse, and system overloads, ensuring a stable and secure API environment for all users.

11.4 Data Retention Schedule

This data retention schedule explains how long Mitel keeps your information.

Table 317: Workflow Retention Period

Workflow Content	Duration	Whether the customer can delete records?
Customer Data e.g. contact details (name, address, phone number, email); call details information (i.e. caller ID, name, time, duration, call routing /menu selections); transcripts etc.	30 days	No
All data collected through the execution of a workflow including data extracted from third party integrations	30 days	No
Configuration Data e.g. Integrations including API keys and credentials	Retained until deleted by user or account is deleted.	Yes
Logs (anonymized and masked)	Retained indefinitely	No

Table 318: Workflow Storage Locations

Region	Storage Location
Europe / European Economic Area (EEA)	Europe
United States	United States
Canada	United States
Australia / New Zealand	Australia

Workflow contents are processed in accordance with Mitel's Terms of Service available at <https://www.mitel.com/legal/mitel-cloud-services-terms-and-conditions>. Customer Data is processed and stored within the region where the Customer account is created. Locations set out herein are default locations but are not absolute. Upon termination of services, subject to any legal, regulatory, and/or operational requirements, Mitel will delete all workflow configuration data within 90 days of termination. Customer acknowledges and agrees that Mitel may make changes to this data retention schedule from time to time.

11.5 Compliance

The CloudLink platform has undergone a SOC 2 Type 2 audit and received an attestation report confirming that its controls were suitably designed and operated effectively over the audit period. For customers subject to the Health Insurance Portability and Accountability Act (HIPAA), CloudLink provides features and safeguards that Customers may use to support their HIPAA compliance. CloudLink is audited against the Security Standards in the HIPAA Security Rule and the HITECH Breach Notification Requirements. However, it's important to note that compliance with HIPAA ultimately depends on how each customer configures and uses the platform. A summary of the audit reports is available upon written request.

This chapter contains the following sections:

- [Variable Scope](#)
- [Flow Expression Calculations](#)
- [Handling a Space in a JSON Parameter](#)
- [Variables are not being replaced with the correct value](#)
- [String Replacement Syntax](#)
- [Integrations](#)

This chapter describes different issues encountered while designing and running a flow.

12.1 Variable Scope

A flow calculates a context for each action that is executed within it. This context includes all the variables (local, global) and previously run action parameters that the action has access to throughout the execution. This action can use these values within its context to make decisions on how the execution of flow will work or assign values within its context.

Limitations

An action run outside a condition or action branch will not have access to action parameters that were run within the branch.

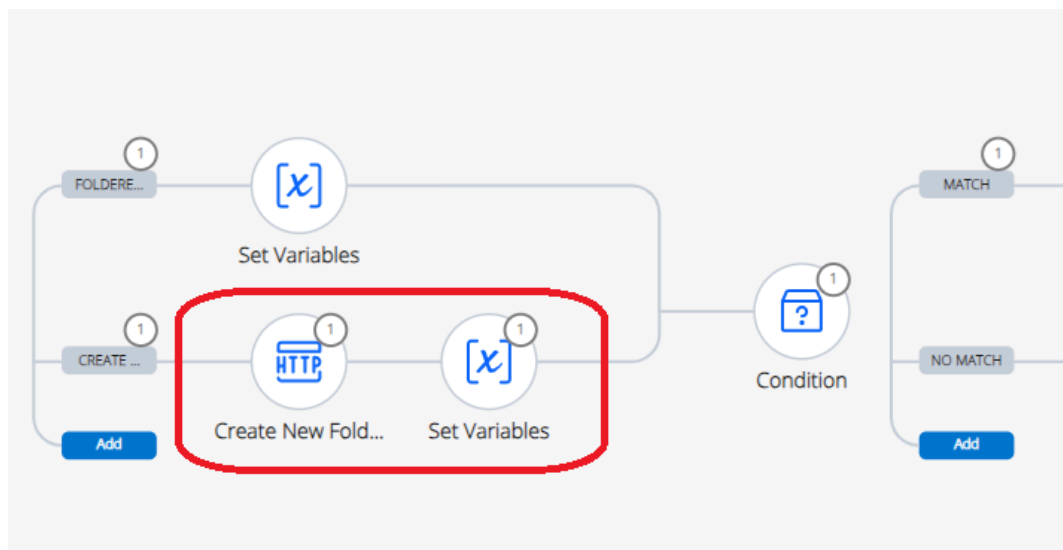


Figure 184: Sample Flow

In the case above, all actions that run beyond the 'Create New Folder' and 'Set Variables' actions and are not part of that branch, will not have access to their throughputs and outputs.

Solutions

There are two possible solutions that can be configured.

1. Create a Local Variable against the flow and assign that variable using a Set Variables action to the throughput/output values of the actions. You can access the local variable from other parts of the flow. For the detailed information, refer to the [Using Variables](#) section.
2. Actions that require access to the throughputs/outputs of these actions must reside in the same branch.

An action does not have access to inputs for previously run actions, only throughputs and outputs.

12.2 Flow Expression Calculations

Evaluating a Script expression that does not match any variables

Problem

A built-in Workflow function allows you to calculate expressions based on the values of variables, throughputs, and outputs of activities that have previously run in a flow. The admin user can select 'Script' as the expression type when assigning a value to an activity argument or a variable. An example is provided below:

Figure 185: Script Expression

When the workflow engine calculates these values, it uses a third-party library called Filtrex to evaluate the value replacement and mathematical calculations. One of the side effects of the library is the scenario where an expression is entered and calculated but nothing has been found to replace the value with. An example of this would be the following:

```
Object = {
  prop1: 'this is my first property'
```

```
}
Expression = 'Object.prop2'
```

Solution

In this scenario, prop2 is not defined within Object.

When the flow runs, the returned value of this expression will be 'Object.prop2'.

This may seem counter intuitive for administrators who have a developer background, as other software languages equate this invalid expression with null.

Using synchronous webhooks to return custom payload

Problem

When using a webhook to return a customized body, if your output name is Body, that will override the default payload as shown below:

```
{
  "example": "value"
}
```

Solution

If you would like to return a custom value instead of the default payload, you can't use the reserved output name **"Body"**. Use another name for the output, for example, **"Response"**.

```
{
  "workerId": "f4b7bbd3-3a15-498c-96a1-90d7d11784ca",
  "outputs": {
    "Response": {
      "example": "testvalue"
    }
  },
  "_links": {
    "self": "/2017-09-01/activities/b65d53f0-bb30-447b-b48f-866800591e42/workers/f4b7bbd3-3a15-498c-96a1-90d7d11784ca"
  }
}
```

My flow is not resuming after the bookmark activity

1. Workflow being executed did not have a worker id, for example, <https://workflow.us.dev.api.mitel.io/2017-09-01/webhooks/accounts/{{accountId}}/activities/{{flowId}}/workers/{{workerId}}>
 - workerId - can be found in the response payload of the initial call
 - accountId / flowId - refer to the [Debug](#) section.
2. The workflow has already been completed. To verify, refer to the [Running](#) section.

12.3 Handling a Space in a JSON Parameter

Problem

The parameter matching fails when you add a space in a JSON parameter. For example, `SystemEventBody.visitor.additional.Car Reg`

Solution

To use a space in a parameter, you will need to use a JavaScript activity. For example, `event.SystemEventBody.visitor.additional['Car Reg']`

12.4 Variables are not being replaced with the correct value

Problem

The expression being evaluated does not exist.

`myData` is not found, this will evaluate to `Response = "myData"`

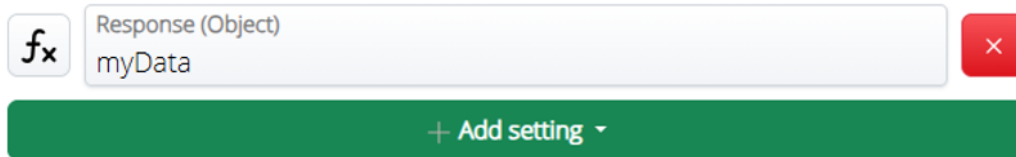


Figure 186: Response Issue

Solution

To investigate this issue, perform the following steps:

1. Add a **log** action to output `myData`.
2. Make sure that `myData` exists within the flow definition/scope.
3. Expressions are case-sensitive.
4. Debug the flow to view the **log** action and validate that the variable has the expected value. Refer to the **Debug** section for instructions on how to debug a flow.

12.5 String Replacement Syntax

Problem

While using string replacement syntax `{{variable1}}`, my variable is not being replaced.

Solution

Make sure the following parameters are properly configured:

1. The Outcome should be of type string
2. The expression should be type Literal.

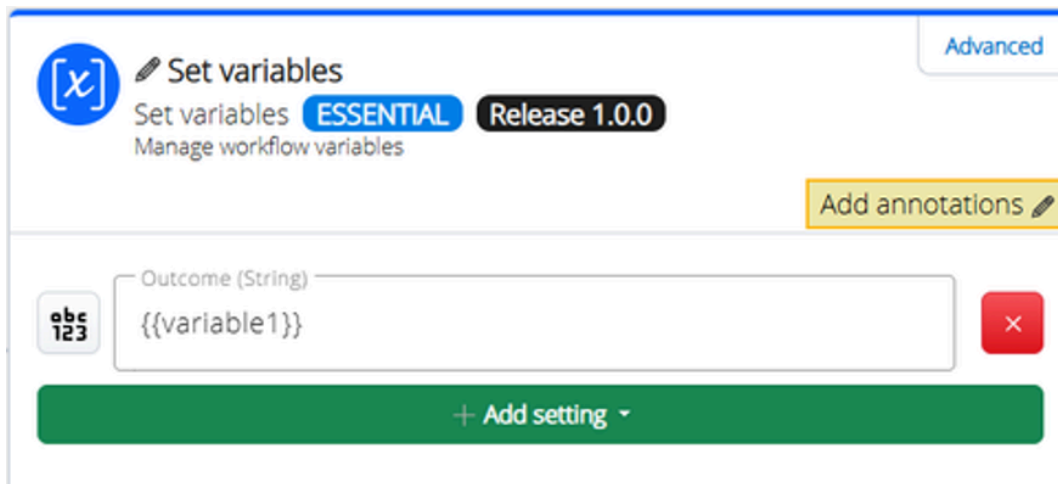


Figure 187: Set Variables

3. The variable is available in scope.
4. Variables are case-sensitive.

12.6 Integrations

Integrations in the activity wheel are not displaying

This occurs because no integrations have been configured under the **Integrations Hub** tab.



Figure 188: Activity Wheel Issue

Solution

Refer to the [Integrations](#) section to set up new connections.

How do you view only Voice flows within Workflow Studio?

Solution

To view only flows that are triggered by a phone call,

1. In the **Flows** screen, click **Add filter**.
2. Select the **By Trigger** option from the list.
3. In the **Filter by Trigger(s)** popup screen, search and select the trigger name as "**Endpoint Ringing**".
4. Click **OK**.

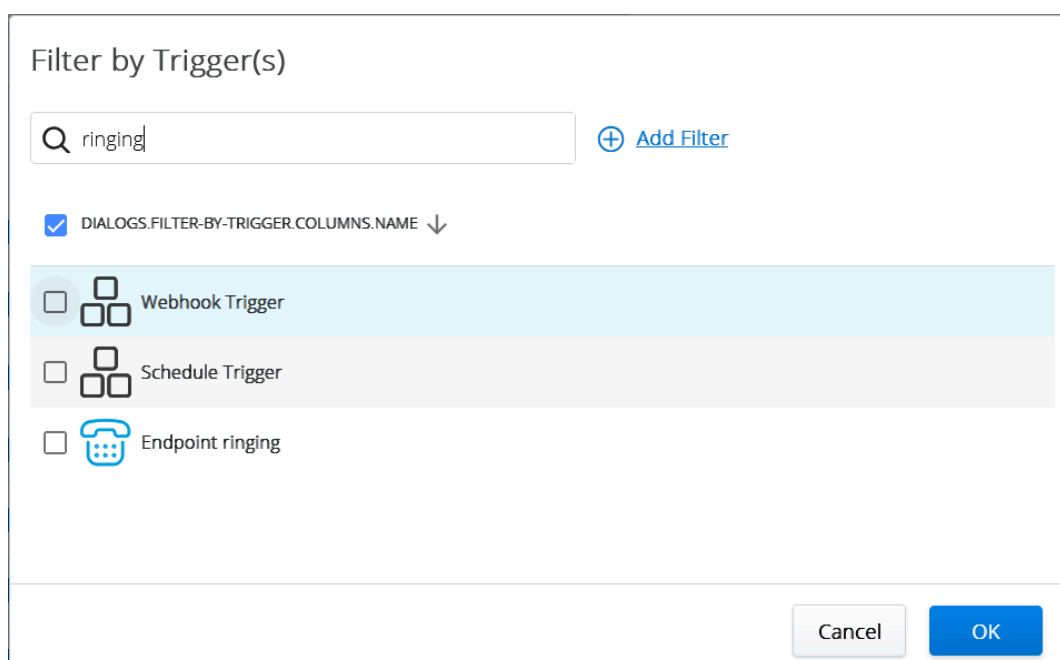


Figure 189: Filter by Trigger(s)

Integration action is not working/always failing

Solution

This can be caused by one of the following reasons:

1. The connection to the integration is no longer valid. Please check the **Integration** view.

Integration Step 1 **Integration Step 2** **Integration Step 3**

Add new

OAuth 2

This integration supports connections using OAuth2. This may require registering a callback URL with the service provider. The following URL can be used during configuration:

`https://workflow.us.dev.api.mitel.io/2017-09-01/integrations/microsoft/sso`

Microsoft Graph API OAuth2 Authorization Code 3
⚠ This integration is not functioning. ⚙️ ☐

Microsoft Graph API OAuth2 Authorization Code 1
⚠ This integration is not functioning. ⚙️ ☐

Incoming Email Subscription
✅ Connected ⚙️ ☐

Cancel **Back**

Figure 190: Invalid Connections

2. Verify that the connection exists.
3. Verify that the permissions are correct using the **Validate** button on the connection.

Configure this connection

Name *
Incoming Email Subscription

tenant *
Placeholder text

client_id *
Placeholder text

client_secret *
●●●●●●●●

scope *
https://graph.microsoft.com/.default

Validate **Refresh** **Save**

Figure 191: Validate Connection

