UiPath Communications Mining Dispatcher Project

How to Use the Communications Mining Dispatcher Project

Introduction

This document provides detailed, step-by-step instructions for using the **Communications Mining Dispatcher Project**. The project is designed to consume communications streams via API calls, extract communications, and create queue items for further processing.

Prerequisites

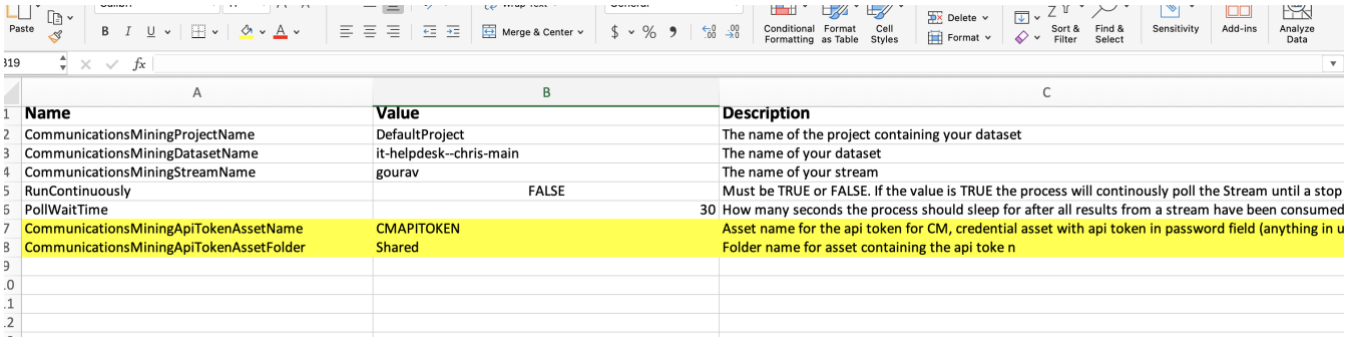
UiPath Studio
Orchestrator Access (for Assets and Queues)
Communications Mining API credentials (Base URL, Authorization Token)

Project Structure Overview

Component	Purpose
Main.xaml	Main entry point of the Dispatcher
Framework/InitAllSettings.xaml	Initializes settings and configurations
Framework/AdvanceStream.xaml	Advances the stream to the next communication
Framework/GetStreamResults.xaml	Retrieves current communication items from the stream
Framework/GetNextCommunication.xaml	Extracts next available communication item
Framework/CreateQueueItem.xaml	Creates queue items in Orchestrator
Data/Config.xlsx	Holds configuration values like API URLs and credentials

Step 1: Configure Project Settings

1. Open the Data/Config.xlsx file.
2. Fill in the following details under the **Settings** and **Assets** sheet:
    - .

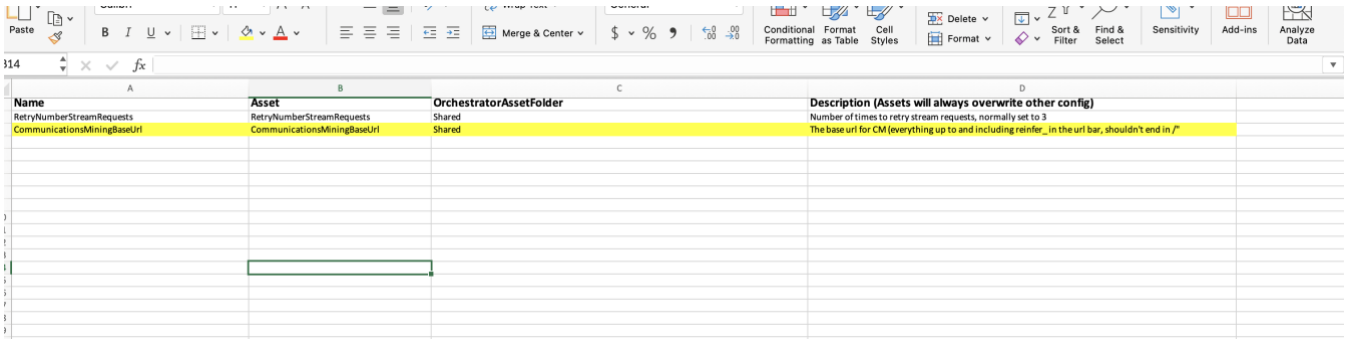


Name	Value	Description
CommunicationsMiningProjectName	DefaultProject	The name of the project containing your dataset
CommunicationsMiningDatasetName	it-helpdesk--chris-main	The name of your dataset
CommunicationsMiningStreamName	gourav	The name of your stream
RunContinuously	FALSE	Must be TRUE or FALSE. If the value is TRUE the process will continously poll the Stream until a stop
PollWaitTime	30	How many seconds the process should sleep for after all results from a stream have been consumed
CommunicationsMiningApiTokenAssetName	CMAPITOKEN	Asset name for the api token for CM, credential asset with api token in password field (anything in u
CommunicationsMiningApiTokenAssetFolder	Shared	Folder name for asset containing the api toke n

Step 2: Set Up Orchestrator Assets

Create Assets in Orchestrator for below

- CommunicationsMiningApiTokenAssetName
- CommunicationsMiningApiTokenAssetFolder
- CommunicationsMiningBaseUrl
- Update Config.xlsx to fetch from Assets.



Name	Asset	OrchestratorAssetFolder	Description (Assets will always overwrite other config)
RetryNumberStreamRequests	RetryNumberStreamRequests	Shared	Number of times to retry stream requests, normally set to 3
CommunicationsMiningBaseUrl	CommunicationsMiningBaseUrl	Shared	The base url for CM (everything up to and including reinfer_ in the url bar, shouldn't end in /"

Step 3: Review Orchestrator Queues

Make sure the **DestinationQueue** exists in Orchestrator:

- Go to **Orchestrator → Queues → Create a new Queue** if needed.

Step 4: Open Project in UiPath Studio

1. Open **UiPath Studio**.
2. Click on **Open a Local Project**.
3. Navigate to and open the folder **CommunicationsMiningDispatcherDirectToApi**.
4. Open Main.xaml.

Step 5: Understand the Execution Flow

1. **InitAllSettings.xaml** – Loads Config values.
2. **AdvanceStream.xaml** – Moves to the next communication in the stream.
3. **GetStreamResults.xaml** – Retrieves communication items.
4. **GetNextCommunication.xaml** – Picks the next available communication.
5. **CreateQueueItem.xaml** – Creates a Queue Item in Orchestrator.
6. **Repeat** – Continues until no more communications are left.

Step 6: Run the Dispatcher

- Click **Run** (or **Debug**) in UiPath Studio.
- Monitor logs for activities like “Stream Advanced”, “Stream Results Fetched”, “Queue Item Created”.

Step 7: Monitor Orchestrator

- Navigate to **Orchestrator**.
- Check **Queues → DestinationQueue** to see new items being created.