

Genesys Cloud powered by UiPath

IVR (“Interactive Voice Response”) systems are often the first point of contact for customers as part of an enterprise’s contact care system. Unfortunately, IVRs often fall short on the promise of providing “self-service” solutions for customers as IVRs are only capable of connecting to systems that are accessible via API, which even when available can be expensive to implement such solutions.

With UiPath’s Robotic Process Automation (RPA) platform, [Genesys Cloud](#) can be connected to any system via RPA, drastically expanding IVR fulfillment capabilities to gather information and perform actions across a plethora of systems from on-prem mainframes to cloud-based web services. Increasing the success rate of customer requests being fulfilled by the self-service IVR frees your contact center agents up to work on more complex customer issues. The results? Improved customer and employee experiences, enhanced accuracy, reduced Average Handling Time (“AHT”), and acceleration of your digital transformation initiatives, resulting in a rapid return on investment.

The UiPath Enterprise Platform connects Genesys Cloud to any system, including those not previously accessible via Genesys Cloud. This enables a zero-touch, fully automated contact center solution and complete customer self-service, freeing up agents to work on more complex customer issues. This integration services all regions where Genesys Connect is available globally across Americas, EMEA, and APAC.

Key Features

- **Integrate with all applications with UiPath UI automation and native integrations.** UiPath world-class UI automation capabilities enables automation of any application, browser, or virtual environment. Leveraging UI automation or native integrations, UiPath integrates with any system that your contact center relies on, including systems previously not accessible by Genesys Cloud
- **Seamlessly transfer requests from self-service to contact center agent with human-in-the-loop.** UiPath robots can be leveraged for a seamless handoff from a self-service IVR channel to a live agent, helping agents solve customer problems faster, reduce AHT, and improve satisfaction.
- **Quickly scale to adjust to business needs.** From small automation projects to company-wide extensive scopes, UiPath Orchestrator and UiPath Robots can quickly scale alongside Genesys Cloud to meet any enterprise need for normal business cycles or unplanned spikes in call volume in your contact center.

Benefits

- **Empower agents to focus on complex issues:** the UiPath integration with Genesys Cloud increases request fulfillment success rates in the self-service IVR, freeing contact center agents to work on more complex customer issues.

- **Realize rapid ROI:** Streamlined integration capabilities, easy-to-use customizable automation templates and an intuitive UI in UiPath Studio ensure fast deployment and a quick return on investment (ROI) alongside your Genesys Cloud instance.
- **Improve customer & employee satisfaction:** With UiPath and Genesys Cloud, customers and employee requests are resolved quickly, thereby boosting CSAT and NPS.

Demo: Fulfilling IVR requests with UiPath automation

Aileen is a customer of Meeple Telco and wants to know the current balance for her mobile phone bill. She makes a voice call to her Meeple Telco customer service number and the IVR flow queries the back-end for her last bill date and total. Aileen is thrilled that she was able to answer her question in under 30 seconds and without having to talk to anyone!

Deployment Steps

UiPath has provided two different flavors of this integration:

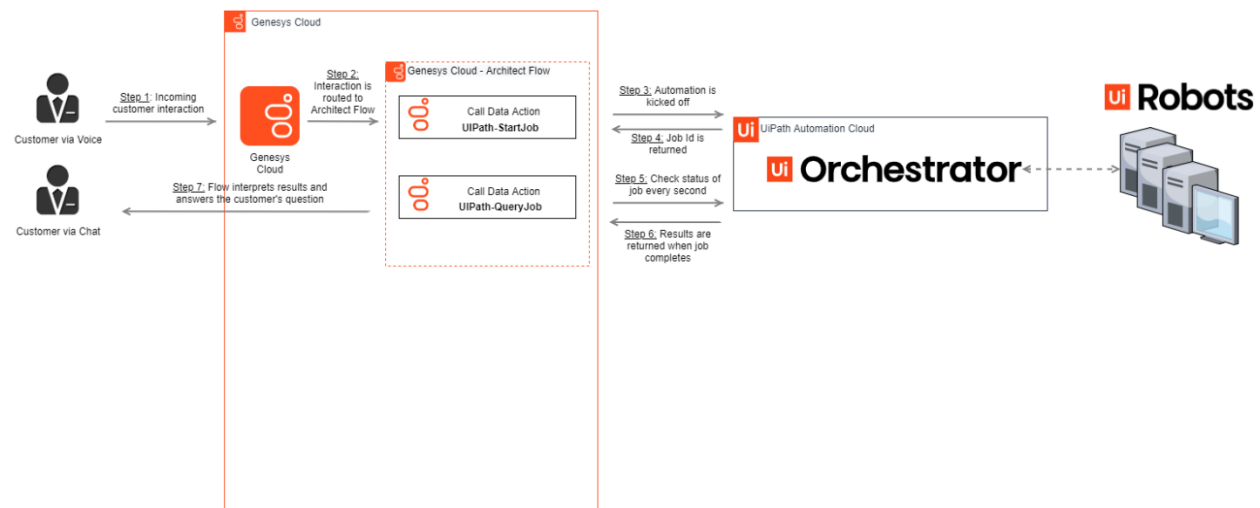
- Calling UiPath directly via Genesys Cloud Data Actions
- Calling UiPath via AWS Lambda

Directions for setting up both integrations are below.

Calling UiPath directly

Genesys Cloud powered by UiPath

Fulfill Genesys Cloud IVR requests with unattended UiPath automation



Step 1. Create & configure your UiPath Cloud instance

1. Create a new UiPath cloud instance, or use an existing instance

2. Download the sample processes [in this repo](#), or use your own processes
 - BillLookup - an unattended automation that takes a phone number as input and returns the last monthly bill details
3. Publish the processes from Studio to Orchestrator - [HowTo](#)
4. Deploy the processes - [HowTo](#)
5. Generate a user key and client ID for your cloud instance - [HowTo](#)
6. Note down the following information as it will be needed in later steps:
 - User Key
 - Client Id
 - Account logical name
 - Tenant logical name

Step 2. Setup the Genesys Cloud Web Services Data Action integration

1. Navigate to the Interactions/Integrations page, i.e. <https://apps.mypurecloud.com/admin/#/integrations/apps>
2. Add a **Web Services Data Action** integration to your instance per the [Genesys Cloud documentation](#).
3. Name the integration, i.e. UiPath Web Services Data Actions
4. Switch to the Configuration tab, then the Credentials tab, and then click the Configure button.
5. Choose **User Defined (OAuth)** and add 3 credential fields:

Field Name	Value
authUrl	https://account.uipath.com/oauth/token
clientId	The Client Id value from step 1.6
userKey	The UserKey value from step 1.6

Configure Credentials

Credential Type

User Defined (OAuth) ▼

Field Name

Value

authUrl



https://account.uipath.com/oauth



clientId

8DEv1AMNXczW3y4U15LL3jYf62j



userKey

A0bbDcE9FQnrZOSOd-Pc-Y0s4AI



+ Add Credential Field

OK

Cancel

6. Click the OK button on the Configure Credentials dialog
7. Click the Save button on the Configuration page
8. Mark the Integration as active

Step 3. Configure the UiPath Web Services Data Actions (Auth) action

1. Navigate to the Integrations/Actions page, i.e. <https://apps.mypurecloud.com/admin/#/integrations/actions>
2. Find the “Custom Auth” action starting with the name you gave the integration in step 2.3, i.e. UiPath Web Services Data Actions (Auth)
3. Click the ellipsis for the action and choose Edit Action
4. Switch to the Setup tab, then the Configuration tab.
5. Click the Viewing toggle control at the bottom of the screen to change it to Draft mode.
6. Change the Request URL template to **\${credentials.authUrl}**
7. Remove the Authorization header
8. Change the Content-Type header to **application/json**
9. Change the Request Body Template to the following JSON blob

```
{
  "grant_type": "refresh_token",
  "client_id": "${credentials.clientId}",
  "refresh_token": "${credentials.userKey}"
}
```

```
}
```

Request URL Template

`${credentials.authUrl}`

The URL to the desired API. Will start with '/api/' if your action connects back to Genesys Cloud.

Headers

Content-Type

application/json

[+ Add Header](#)

Request Body Template

```
{
  "grant_type": "refresh_token",
  "client_id": "${credentials.clientId}",
  "refresh_token": "${credentials.userKey}"
}
```

10. Click the Save Draft button
11. Switch to the Test tab and click the Run Action button. This should return “Action successfully run”. If it returns an error, please double check you followed the above steps correctly.
12. Click the Save & Publish button and confirm the publish on the resulting dialog

Step 4. Import and configure the UiPath StartJob Data Action

1. Navigate to the Integrations/Actions page, i.e.
<https://apps.mypurecloud.com/admin/#/integrations/actions>
2. Import the action, CustomActions\UiPath-StartJob.custom, located in the downloaded ZIP from this Connect Marketplace listing.
3. Click on the integration you created in step 2.3 and click the Import Action button
4. Click the Save & Publish button and confirm the publish on the resulting dialog

Step 5. Import and configure the UiPath QueryJob Data Action

1. Navigate to the Integrations/Actions page, i.e.
<https://apps.mypurecloud.com/admin/#/integrations/actions>
2. Import the action, CustomActions\UiPath-QueryJob.custom, located in the downloaded ZIP from this Connect Marketplace listing.
3. Click on the integration you created in step 2.3 and click the Import Action button
4. Click on the Setup tab
5. Change the QueryJob data action to match the output of the job you intend to call.

1. Update the Output Contract on the Contracts Tab to include named properties for each output argument of the UiPath process
2. Update the Response on the Configuration Tab to map the outputs to these properties.
6. Click the Save & Publish button and confirm the publish on the resulting dialog

Step 6. Create the Architect flows in Genesys Cloud

1. Open the Genesys Cloud Architect page at <https://apps.myCloud.com/architect/#/inboundcall/flows>
2. Import the sample flow file, **Sample Flow calling UiPath via Data Action.i3InboundFlow**, located in the downloaded ZIP from this Connect Marketplace listing, as a new Flow in Architect per the [Genesys Cloud documentation](#).
3. Navigate to the Get Bill action.
4. Update the flow to point to the newly created data actions
 1. Change the category to the integration you created in Step 2.3
 2. Change the data action to the Start Job data action you created in Step 4.2.
NOTE: If it does not appear in the list, make sure you published it in Step 5.2.4.
 3. Provide the inputs to start the job (Task #15)
 - releaseKey - provide the GUID for the process to run, [see below](#)
 - accountLogicalName –you retrieved this value in step 1.6
 - tenantLogicalName –you retrieved this value in step 1.6
 - folderId - provide the ID for the Orchestrator folder, [see below](#)
 - [Optional] inputArguments – provide the inputs to the UiPath process as a JSON string. For example, ‘{“arg1”: “value1”, “arg2”:0}’
 4. Store the success output, job key as a variable, i.e. Task.jobKey (Task #15)
 5. Provide the inputs to query the job (Task #23)
 - jobKey – this should be the variable output from Task #15 (Task.jobKey)
 - accountLogicalName –you retrieved this value in step 1.6
 - tenantLogicalName –you retrieved this value in step 1.6
 - folderId - provide the ID for the Orchestrator folder, [see below](#)
 6. [Optional] Store the success outputs as variables to be used elsewhere in the flow.
5. Publish the completed flow per the [Genesys Cloud documentation](#).

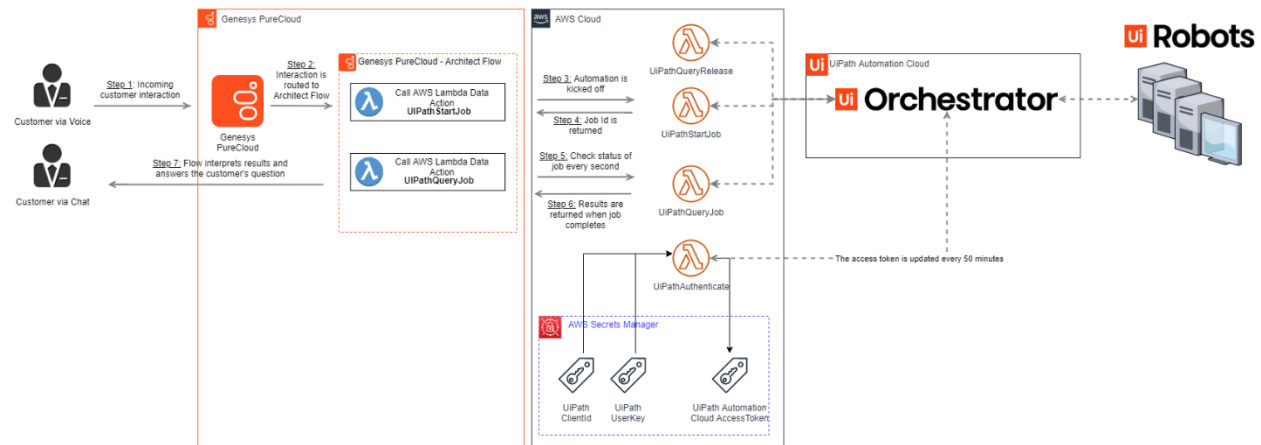
Step 7. Configure & test your flows

1. To Test the flow, open up the phone dialer and call the flow directly. The format to dial is Name@localhost where Name is the name of the flow with spaces replaced by “%20”. For example, I’ve called my flow “UiPath Demo” so I would call “UiPath%20Demo@localhost” to test the flow.

Calling UiPath via AWS Lambda

Genesys PureCloud powered by UiPath

Fulfill Genesys PureCloud IVR requests with unattended UiPath automation



Step 1. Create & configure your UiPath Cloud instance

6. Create a new UiPath cloud instance, or use an existing instance
7. Download the sample processes [in this repo](#), or use your own processes
 - BillLookup - an unattended automation that takes a phone number as input and returns the last monthly bill details
7. Publish the processes from Studio to Orchestrator - [HowTo](#)
8. Deploy the processes - [HowTo](#)
9. Generate a user key and client ID for your cloud instance - [HowTo](#)
10. Note down the following information as it will be needed in later steps:
 - User Key
 - Client Id
 - Account logical name
 - Tenant logical name

Step 2. Prepare an Amazon Connect instance

1. Sign in to your AWS account at <https://aws.amazon.com> with an AWS Identity and Access Management (IAM) user role that has the necessary permissions.
2. If you don't already have an Amazon Connect instance, see the [AWS documentation](#) for information on how to create an Amazon Connect instance.

Step 3. Run the CloudFormation Template

To make things easy to deploy, we've hosted the default template in UiPath's S3 bucket so all you have to do is create a new stack with it. Alternatively, you can customize the [template in this repo](#).

1. Open the [AWS CloudFormation Stack creator](#)

2. Set the region to the same region as your Amazon Connect instance. *NOTE: The AWS Lambda functions must be in the same region as your Cloud organization. For more information, see [AWS regions for Cloud deployment](#).*
3. Fill in the form with the pertinent data
 - Stack Name - provide a name for your stack
 - EXAMPLE: UiPath-AmazonConnect
 - User Key - provide the UserKey name from step 1.6 above
 - EXAMPLE: GHFabdDEfGHjklm0N1QrS2t-3u4vw_XyZ5AB-6cde7Fg
 - Account Logical Name - provide the account name from step 1.6 above
 - EXAMPLE: MyAccountName
 - Tenant Logical Name - provide the tenant name from step 1.6 above
 - EXAMPLE: MyOrchestrator
 - Client Id - provide the clientId name from step 1.6 above
 - EXAMPLE: 8DEv1AMNXczW3y4U15LL3jYf62jK93n5
 - S3 Bucket - *NOTE: this must be all lowercase*
 - EXAMPLE: uipath-amazonconnect-contactflows
4. Continue pressing **Next** for 2 screens.
5. Check the **I acknowledge...** checkbox and finally click **Create Stack**. It should take less than 2 minutes for all the resources to be created.
6. When the stack creation is complete, navigate to the Resources tab and click on the Physical ID of the following Lambda functions:
 - UiPathQueryJob
 - UiPathStartJob
7. Note down the ARNs from the top right of the page. This will be needed in Step 3.1 below.

Here's an example of the completed form:

The screenshot shows the AWS CloudFormation console interface. The top navigation bar includes the AWS logo, 'Services', 'Resource Groups', and user information. The left sidebar shows a progress bar with four steps: 'Step 1: Specify template', 'Step 2: Specify stack details' (active), 'Step 3: Configure stack options', and 'Step 4: Review'. The main content area is titled 'Specify stack details' and contains several input fields for configuring the stack.

Stack name

Stack name

 Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

Orchestrator URLs

Automation Cloud URL
 URL for the UiPath Automation Cloud environment. This should not be changed.

Account Authentication URL
 URL for the authentication endpoint. This should not be changed.

Orchestrator API Access

User Key
 Unique login key to be used with APIs. This is found on the UiPath API Access page.

Account Logical Name
 Your automation cloud account name. This is found on the UiPath API Access page.

Tenant Logical Name
 The orchestrator service's name. This is found on the UiPath API Access page.

Client Id
 Id string specific to the Orchestrator application itself. This is found on the UiPath API Access page.

Contact Flows S3 bucket details

S3 Bucket for saving Contact Flows
 Desired name for the S3 bucket we will create and place sample contact flows in.

At the bottom right, there are three buttons: 'Cancel', 'Previous', and 'Next'.

Step 4. Grant Genesys Cloud access to call the AWS Lambda functions

1. Create an IAM resource to invoke AWS Lambda functions per the [Genesys Cloud documentation](#).

Step 5. Create the AWS Lambda Data Actions in Genesys Cloud

1. Add an AWS Lambda data action integration to your instance per the [Genesys Cloud documentation](#).
2. Create the AWS Lambda data action for both of the AWS Lambda functions by importing the JSON located in the downloaded ZIP from this Connect Marketplace listing per the [Genesys Cloud documentation](#). Be sure to select the integration you created in step 4.2 above.

- UiPathQueryJob – CustomActions\UiPath-QueryJob.custom
 - UiPathStartJob - CustomActions\UiPath-StartJob.custom
3. After importing the data actions, update the ARN to the point to the newly created AWS Lambda function from Step 3.6 above.
 1. Click on the Setup Tab of the newly created action
 2. Click on the Configuration Tab
 3. Paste the ARN from Step 3.6 above into the “Request URL Template” textbox.
 4. Click Save & Publish
 4. [Optional] Change the UiPath-QueryJob data action to match the output of the job you intend to call.
 1. Update the Output Contract on the Contracts Tab to include named properties for each output argument of the UiPath process
 2. Update the Response on the Configuration Tab to map the outputs to these properties.

Step 6. Create the Architect flows in Genesys Cloud

4. Open the Genesys Cloud Architect page at <https://apps.myCloud.com/architect/#/inboundcall/flows>
5. Import the sample flow located in the downloaded ZIP from this Connect Marketplace listing a new Flow in Architect per the [Genesys Cloud documentation](#).
6. Navigate to the Get Bill action.
7. Update the flow to point to the newly created data actions
 1. Change the category to the integration you created in Step 5.1
 2. Change the data action to the data action you created in Step 5.2. *NOTE: If it does not appear in the list, make sure you published it in Step 5.2.4.*
 3. Change the releaseKey and folderId inputs
8. #15 Start Job
 1. releaseKey - provide the GUID for the process to run, [see below](#)
 2. folderId - provide the ID for the Orchestrator folder, [see below](#)
9. #23 Query Job
 1. folderId - provide the ID for the Orchestrator folder, [see below](#)
10. Publish the completed flow per the [Genesys Cloud documentation](#).

Step 7. Configure & test your flows

2. To Test the flow, open up the phone dialer and call the flow directly. The format to dial is Name@localhost where Name is the name of the flow with spaces replaced by “%20”. For example, I’ve called my flow “UiPath Demo” so I would call “UiPath%20Demo@localhost” to test the flow.

Obtaining the folder id

The Folder Id represents the Orchestrator folder your processes are stored in.

1. Open up a browser and sign into your [Cloud Orchestrator instance](#)
2. Navigate to <https://cloud.uipath.com/ACCOUNT/TENANT/odata/Folders> where ACCOUNT is your account name and tenant is your tenant name (see step 2.6 above). This will return a JSON blob like the following:

```
{
  "@odata.context": "https://cloud.uipath.com/ACCOUNT/TENANT/odata/$metadata#Folders",
  "@odata.count": 2,
  "value": [
    {
      "DisplayName": "Default",
      "FullyQualifiedName": "Default",
      "FullyQualifiedNameOrderable": "Default",
      "Description": null,
      "ProvisionType": "Manual",
      "PermissionModel": "InheritFromTenant",
      "ParentId": null,
      "IsActive": true,
      "Id": 12345
    },
    {
      "DisplayName": "HR",
      "FullyQualifiedName": "HR",
      "FullyQualifiedNameOrderable": "HR",
      "Description": null,
      "ProvisionType": "Manual",
      "PermissionModel": "InheritFromTenant",
      "ParentId": null,
      "IsActive": true,
      "Id": 67891
    }
  ]
}
```

3. Take the Id field for the folder your processes are stored in. The DisplayName is likely 'Default' like in the example above.

Obtaining a release key

We have provided a helper Lambda function, `UiPathQueryRelease`, for converting a process name into the release key GUID. You can [manually run the lambda from the Lambda Console](#) to get the value you need to provide in the contact flow configuration in step 4.3. The simplest approach is to create a new test event based on the *connect-contact-flow-*

event event template. Below is an example of the input you need to pass the lambda. Note that the process name passed in must match the name found in Orchestrator.

Input Format

```
{
  "Name": "ContactFlowEvent",
  "Details": {
    "ContactData": {
      "ContactId": "NONE"
    },
    "Parameters": {
      "folderId": "YOUR_FOLDER_ID",
      "processName": "PROCESSNAME"
    }
  }
}
```

Input Example

```
{
  "Name": "ContactFlowEvent",
  "Details": {
    "ContactData": {
      "ContactId": "NONE"
    },
    "Parameters": {
      "folderId": "12345",
      "processName": "BillLookup_MyEnvironment"
    }
  }
}
```

To post feedback, submit feature ideas, or report bugs, use the Issues section of this GitHub repo.