Setting up the “Connect to a Printer” Skill for an AtBot Enterprise Demo

# View a video of this AtBot Skill in action

* [in Microsoft Teams](https://youtu.be/4JSfjvT0Bpk)
* [in SharePoint Online](https://youtu.be/wAPcafOL_tg)

# Set up your bot

If you haven’t already, be sure to create and configure an AtBot by following steps 1 through 3 in the [Enterprise Jump Start Guide](https://tinyurl.com/AtBotJumpStart). This document is your go-to guide for getting your enterprise AtBots up and running and you should keep it handy whenever you’re building out AtBots.

## Skill overview

This AtBot Skill mimics an IT support chatbot helping a user with connecting to a network printer. The example is useful because even in our modern world, people still need to print and printing is sometimes a confounding process. The bot doesn’t only help the user connect to a printer; it also suggests a printer to the user based on where they work (or where they want to print) and which type of printer they need.

As built, this Skill uses a SharePoint list with a number of printers to query and provide dynamic responses based on the user selections. This portion of the demo is optional. You can replace some steps with pre-determined responses for a less dynamic—though just as impactful—demo as long as you do not plan to discuss the data source of the printer list.

This Skill is intended to connect to an IT trouble ticketing system. Microsoft Flow has built-in connections to popular ticketing tools like Zendesk, Service Now, and Freshdesk, presuming you have a [premium Flow plan](https://us.flow.microsoft.com/en-us/pricing/) for the bot’s service account in Flow. If you have demo or live accounts with those systems, take advantage of creating a ticket as the first step of the process, with a ticket subject of “Get user connected to a network printer” or something similar. Close the Skill out by closing the ticket if it was successful or asking for more information and elevating the ticket to a human if the bot was unsuccessful.

This is a reasonably complex Skill and the Flow requires some updates regardless of whether you decide to connect to SharePoint. There are multiple switch cases that will require updates to ensure it runs correctly if you choose not to list information dynamically from SharePoint.

This is a powerful demonstration to IT organizations as they see the real value of automating tier-1 support. Ultimately, most recipients of the demo will likely ask how they are expected to create a Skill for every possible problem that IT could experience. The smart response to this is to suggest the IT team reviews their existing ticketing metrics and automate the top 15 or 25 ticket types to start (each its own Skill) and add more as they have time. Provisioning of Office Groups, SharePoint sites, and licenses for professional software like Adobe Acrobat Pro, Creative Cloud, and others is another excellent use case that resonates with IT teams everywhere.

# Import the LUIS App

The LUIS App provides the natural language processing for the demo. It will be used to kick off the Flow and it will also take in the expense item, the cost, and the date based on the user’s input.

1. Download the LUIS App (JSON file) stored in the [demo resources GitHub](https://github.com/iamatbot/Demo-Skills/tree/master/IT%20Bot/Connect%20to%20a%20Printer/Enterprise%20AtBot%20Demo).
2. Open luis.ai and sign in.
3. In the My Apps listing, click **Import new app** > **Choose app file (JSON format)…** > **upload** the file downloaded in Step 1 of this section > **Train** > **Publish** (to production).
4. Click your name in the top-right corner > **Settings**.
5. Copy your **Authoring Key**. Place it somewhere for later (a text file, a Word document, a task in Outlook, etc.).

# Create the LUIS Intent Vector

The LUIS Intent Vector adds on the LUIS intent by making certain input required. If some input wasn’t provided, the IV will then request the input from the user. In the case of this Skill, the required information is the chrome type of the preferred printer (color or black and white). The bot will need to know this before it can find a printer for the requesting user.

1. Open the AtBot Admin Portal at admin.atbot.io.
2. Click **AI Integrations** > **LUIS Intent Vectors** > **Create Intent Vector**.
3. Enter the LUIS App Region.
4. Enter the copied **Authoring Key** from step B.5.
5. Select the LUIS App (the default name of the provided App is “Connect to Printer”).
6. Select the LUIS Intent (the default name of the provided Intent is “Connect to Printer”).
7. Click **Create Intent Vector**.
8. Add the **Available Entity** (Printer type) as a **Configured Entity** by pressing the plus sign next to the entity in the **Available Entities** column.
9. Configure the entities to look like the LUIS Intent Vector file in the [demo resources GitHub](https://github.com/iamatbot/Demo-Skills/tree/master/IT%20Bot/Connect%20to%20a%20Printer/Enterprise%20AtBot%20Demo). Exact text match is required for **Choices Options**.
10. Click **Update Vector**.

Your Intent Vector will now be available to choose as an intent in the trigger in your Flow.

# Import the Flow

The Flow is the stepped workflow that runs once LUIS triggers it. The Flow will not run until all required input is supplied via the Intent Vector (section C above). As built, the Flow requires a SharePoint list to query a listing of printers; this portion is optional. See section E for details. Some steps below are easier to follow with the help of Figure 1, below.

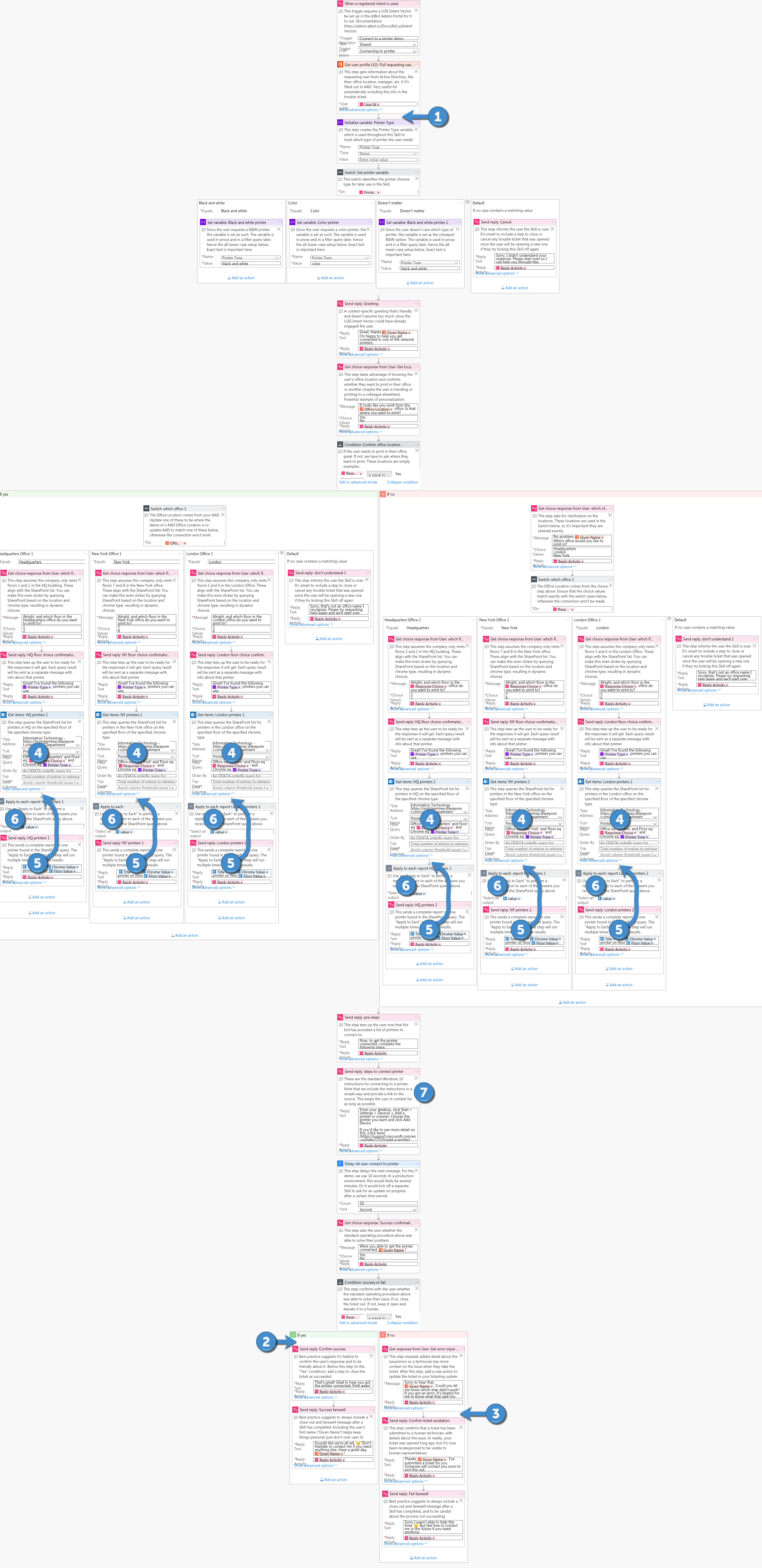
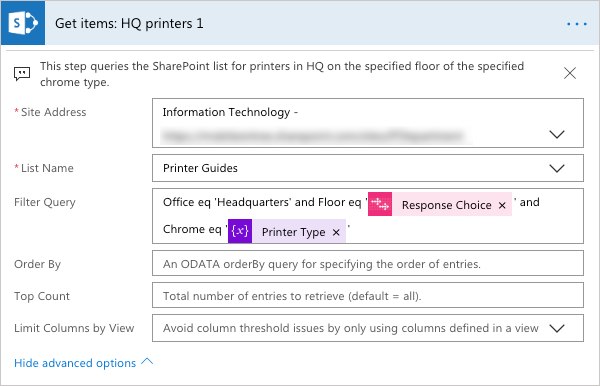


Figure 1: Overview of the Flow. Review the steps below before making any changes referenced above.

Import your Flow:

1. Download the Flow (ZIP file) stored in the [demo resources GitHub](https://github.com/iamatbot/Demo-Skills/tree/master/IT%20Bot/Connect%20to%20a%20Printer/Enterprise%20AtBot%20Demo).
2. Open flow.microsoft.com and sign in.
3. Click **My Flows** > **Import** > **Upload** > Choose your file. Update each of the Resource Types as listed below. Once they are selected, click **Import**. For each Resource Type:
   1. Set *Flow* to “Create as new”.
   2. Set *AtBot Logic Connection* to “Select during import”. If there is no connection listed, click **Create new** > **New connection** > Find **AtBot Logic** and **Add (plus)**.Go back to the Flow tab > **Select during import** > choose your newly created connection > **Save**.
   3. Set *SharePoint Connection* to “Select during import”. If there is no connection listed, click **Create new** > **New connection** > Find **SharePoint** and **Add (plus)**.Go back to the Flow tab > **Select during import** > choose your newly created connection > **Save**.
   4. Set *Office 365 Users Connection* to “Select during import”. If there is no connection listed, click **Create new** > **New connection** > Find **Office 365 Users** and **Add (plus)**.Go back to the Flow tab > **Select during import** > choose your newly created connection > **Save**.
4. Open the imported Flow: **My Flows** > **[Flow name]**.
5. Click the trigger. Ensure **Bot Trigger Type** is set to **Shared** and select the LUIS Intent Vector from earlier in that field.
6. Optional: If you want to connect to an IT ticketing system to track a ticket for this Skill, follow the below sub-steps. If not, skip to step 7.
   1. Add a new step after the **Get user profile (V2): Pull requesting user’s data from AAD** step (label 1 in the Figure 1) that creates a new ticket in the destination. Provide a standard ticket description. Set the ticket to “Open” or “New”. (This step comes after the **Get user profile** because you will likely want the user’s information to automatically populate the ticket.)
   2. In the final **Condition: success or fail**, add a new step in the **Yes** column before the **Send reply: Confirm success** step (label 2 in the Figure 1) to update the ticket and close it. Provide a useful closing update like, “User was successfully supported via AtBot Skill.”
   3. In the final **Condition: success or fail**, add a new step in the **No** column after the **Get response from user: Get error input from user** step (label 3 in the Figure 1) to update the ticket with the user’s response from the previous step and to recategorize the ticket to a category that a human technician would see. Provide a useful update like, “AtBot was unable to assist user in this case and reports the following issues: [Response text from previous step].”
7. Dynamically query a printer listing in SharePoint:
   1. If you do not want to do a dynamic pull from SharePoint, you will have to
      1. delete all of the **Get items** steps (label 4 in Figure 1);
      2. move each **Send reply** step in the **Apply to each** sections above to the **Apply to each** (label 5 in Figure 1);
      3. update each **Send reply** step (label 5 in Figure 1) to replace any dynamics properties from SharePoint to be pre-written (you choose the printer names and other details); and
      4. delete the **Apply to each** steps (label 6 in Figure 1).
   2. If you want to dynamically pull data from a SharePoint list for the demo, see section E for instructions on setting up the list. Update each **Get items** step (label 4 in Figure 1) to select the site and list you created. Ensure your Filter Query (under “Advanced Options” in the step) reflects the information shown below.  
      
8. Optional: In the **Send reply: steps to connect to printer** step (label 7 in Figure 1), feel free to update the instructions and resource link if you prefer something else (perhaps a local standard operating procedure and link to a local knowledge base).

Your Flow, which is now connected to a LUIS intent, is considered an AtBot Skill. We will use “Skill” from now on to represent the combination of the two.

# Optional: Configure your SharePoint list

If you choose to include the SharePoint list referenced in step D.7. above, follow these steps to import an existing SharePoint list that you can then query from your Skill. Generally, a modern SharePoint Online site works best; it doesn’t matter whether you use a Team or Communication Site. If you’re creating from scratch, use a Team site and set the members as anyone that will be using the bot or demoing it.

Once you create your site:

1. Download the list spreadsheet (XLSX) stored in the [demo resources GitHub](https://github.com/iamatbot/Demo-Skills/tree/master/IT%20Bot/Connect%20to%20a%20Printer/Enterprise%20AtBot%20Demo).
2. Open you SharePoint site.
3. You can [import the spreadsheet](https://support.office.com/en-us/article/create-a-list-based-on-a-spreadsheet-380cfeb5-6e14-438e-988a-c2b9bea574fa) as a list if you’re using Internet Explorer.
4. Alternately, you can [create a custom list](https://support.office.com/en-ie/article/create-a-list-in-sharepoint-0d397414-d95f-41eb-addd-5e6eff41b083) and [use Quick Edit](https://support.office.com/en-us/article/edit-list-items-in-sharepoint-online-dac1a1c3-a80b-4082-ba57-715cf613d0f7) to copy and paste each cell from the spreadsheet into the list.
5. Once the list is created, you can update the SharePoint steps referenced in section D.

# Configure your AtBot

1. Open the AtBot Admin Portal at admin.atbot.io.
2. If your bot doesn’t have an assigned category, [create a category](https://admin.atbot.io/Docs/Categories) and [assign it to your bot](https://admin.atbot.io/Docs/EnterpriseBots/BasicConfiguration).
3. [Add your new Skill](https://admin.atbot.io/Docs/Skills) into your category.

# Next steps

1. Add a QnA Maker knowledge base utilizing Chit-Chat for some basic small talk ability by your bot. It makes the experience more natural and entices users to trust the bot more if you’re letting others try it out. Review step 6 in the [Enterprise Jump Start Guide](https://tinyurl.com/AtBotJumpStart) for more info.
2. Ensure you’ve provided licenses for testers/demo-ers and add the bot to Microsoft Teams. All steps can be found in the [Enterprise Jump Start Guide](https://tinyurl.com/AtBotJumpStart).