

PL-600: Power Platform Solution Architect

NOTE: Please refer to the TenantSetup.md file for the steps to set up your lab. You will give students credentials to use for their classroom work. It is VERY important that students use this environment you have created and not their own Azure DevOps. As always for classroom work., it is best for students to work in private browser sessions to avoid any potential conflicts they might have with saved credentials.

- [**Download Latest Student Handbook and AllFiles Content**](#)
- **Are you a MCT?** - Have a look at our [**GitHub User Guide for MCTs**](#)
- **Need to manually build the lab instructions?** - Instructions are available in the [**MicrosoftLearning/Docker-Build**](#) repository

What are we doing?

- To support this course, we will need to make frequent updates to the course content to keep it current with the Azure services used in the course. We are publishing the lab instructions and lab files on GitHub to allow for open contributions between the course authors and MCTs to keep the content current with changes in the Azure platform.
- We hope that this brings a sense of collaboration to the labs like we've never had before - when Azure changes and you find it first during a live delivery, go ahead and make an enhancement right in the lab source. Help your fellow MCTs.

How should I use these files relative to the released MOC files?

- The instructor handbook and PowerPoints are still going to be your primary source for teaching the course content.
- These files on GitHub are designed to be used in conjunction with the student handbook, but are in GitHub as a central repository so MCTs and course authors can have a shared source for the latest lab files.
- It will be recommended that for every delivery, trainers check GitHub for any changes that may have been made to support the latest Azure services, and get the latest files for their delivery.

What about changes to the student handbook?

- We will review the student handbook on a quarterly basis and update through the normal MOC release channels as needed.

How do I contribute?

- Any MCT can submit a pull request to the code or content in the GitHub repro, Microsoft and the course author will triage and include content and lab code changes as needed.
- You can submit bugs, changes, improvement and ideas. Find a new Azure feature before we have? Submit a new demo!

Notes

Classroom Materials

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only.

title: Online Hosted Instructions permalink: index.html layout: home

Content Directory

Hyperlinks to each of the lab exercises and demos are listed below.

Labs

```
{% assign labs = site.pages | where_exp:"page", "page.url contains  
'/Instructions/Labs'" %} | Module | Lab || --- | --- | {% for activity in labs %}|  
{ { activity.lab.module } } | {{ activity.lab.title }}.{% if activity.lab.type %}-  
{{ activity.lab.type }}.{% endif %} | {% endfor %}
```

Demos

```
{% assign demos = site.pages | where_exp:"page", "page.url contains '/Instructions/Demos'" %} | Module | Demo || --- | --- | {% for activity in demos %}| {{ activity.demo.module }} | .{{ activity.demo.title }}. | {% endfor %} Setup PL-600 student tenant =====
```

All delegates, and instructor, share the same tenant. As this is a shared environment, some tasks require a tenant Global Administrator role need to be completed by the instructor prior to the course.

It is recommended to use an online lab provided with an Authorized Lab Hoster (ALH) for this course. Alternatively, you can create trial tenant with a Power Apps trial license.

There are a series of instructions that required to performed to prepare for delivery of the course for both online lab and trial options:

- ALM lab setup: Creates users, groups, and environments
- RPA lab setup: Adds a trial of Power Automate RPA and configures users
- App exercise setup: Import a solution and create test data

[!IMPORTANT] Do not use this on your production live tenants as it creates and delete users and environments.

Location of files

The location of the files for the setup for this course are in GitHub

<https://github.com/MicrosoftLearning/PL-600-Power-Platform-Solution-Architect/tree/master/Allfiles>

Step 1 ♦♦♦ Create a new Trial tenant (if required)

If you are not using an Authorized Lab Hoster lab, you will need to create a Power Apps trial. The steps are contained in the **Power Platform Trial Setup** instructions which can be found in *Allfiles**

Step 2 - ALM Lab Setup

Follow the steps in the ALM lab setup document. This uses a PowerShell script to creates user, security groups, environments, and import a solution.

If you see errors or warnings when running the script, resolve them prior to continuing

[!NOTE] If you have incompatible PowerShell modules already installed you might get an error prompting you to run CleanupOldModules.ps1. Close all PowerShell sessions and run th CleanupOldModules.ps1 script which will remove all PowerShell modules used by the setup script so the setup script can then install the versions it requires.

This should now cycle through creating your users and their environments. This may take several minutes. You will see a command prompt when this is complete. Review the log for any errors you may need to address. You may re-run the setup script to attempt to resolve errors that occurred during the prior run.

If you are using an Authorized Lab Hoster lab, there are 24 users pre-created on the tenant. If you are using a trial tenant, the script creates users that follow a naming convention of labadminXX@yourtenant.onmicrosoft.com; with a password of **test\@word1**. The XX is replaced by the numbered user up to the number of users you provisioned with the script.

Step 3 - RPA Lab Setup

Follow the steps in the RPA lab setup document. In these instructions you will add a trial of Power Automate RPA and configures users.

Step 4 - App Exercise Setup

Follow the steps in the App lab setup document. In these instructions you will import the solution used in the exercise.

Application Lifecycle Management lab setup

Objectives

Create users and environments for students to run the Application Lifecycle Management (ALM) lab

Request grant of parallel pipeline jobs

Microsoft have removed the ability to run pipelines in Azure DevOps for newly created pipelines. You should create your Azure DevOps organization as soon as possible before the course and send an email to azpipelines-freetier@microsoft.com.

Include the following information in the email, your name and the name of the Azure DevOps organization: **Device Management lastnameMMYY**

[!NOTE] See <https://devblogs.microsoft.com/devops/change-in-azure-pipelines-grant-for-private-projects/> for more details

You should consider having the delegates perform the same at the beginning of the course so that they will be granted perallelism in time to complete the lab.

Download files from GitHub

1. In Windows, open Windows PowerShell

Make sure you open PowerShell as an **administrator**. See <https://docs.microsoft.com/powershell/scripting/learn/ps101/01-getting-started#how-do-i-launch-powershell> for how to start PowerShell

[!NOTE] You can use the PowerShell command line or the PowerShell ISE. The setup process is documented to use, and was tested with, the Windows PowerShell ISE application. If you are not familiar with Windows PowerShell ISE, you can find more information at <https://docs.microsoft.com/en-us/powershell/scripting/windows-powershell/ise/introducing-the-windows-powershell-ise>.

1. New-Item -Path "C:\" -Name "LabFiles" -ItemType "directory"
2. Set-Location C:\LabFiles
3. Import-Module -Name BitsTransfer
4. Start-BitsTransfer -Source 'https://github.com/MicrosoftLearning/PL-600-Power-Platform-Solution-Architect/archive/master.zip' -Destination C:\LabFiles
5. Expand-Archive -Path 'C:\LabFiles\master.zip' -DestinationPath 'C:\LabFiles'
6. Move-item -Path "C:\LabFiles\PL-600-Power-Platform-Solution-Architect-master*" -Destination "C:\LabFiles" -confirm:\$false
7. Remove-Item master.zip
8. Remove-Item PL-600-Power-Platform-Solution-Architect-master
9. Set-Location C:\LabFiles\Allfiles
10. dir
11. Verify that the Setup.ps1 and ContosoDeviceOrderManagement_1_0_0_1.zip files are in the location.

Directory:			
Mode	LastWriteTime	Length	Name
-	--	--	.
d----	09/03/2021 16:18		Demos
d----	09/03/2021 16:18		Labs
a----	09/03/2021 16:18	381	CleanupOldModules.ps1
a----	09/03/2021 16:18	115140	ContosoDeviceOrderManagement_1_0_0_1.zip
a----	09/03/2021 16:18	19838	M02-Project Workbook.xlsx
a----	05/01/2021 13:55	861980	Power Platform Trial Setup.docx
a----	10/03/2021 10:24	7106	PowerShell Setup.zip
a----	10/03/2021 10:20	39383	Setup.ps1

PowerShell setup script

1. In Windows, open Windows PowerShell

Make sure you open PowerShell as an **administrator**. See <https://docs.microsoft.com/powershell/scripting/learn/ps101/01-getting-started#how-do-i-launch-powershell> for how to start PowerShell.

2. Run the following commands:

- a. **Set-ExecutionPolicy RemoteSigned**
- b. **Set-PSRepository -Name PSGallery -InstallationPolicy Trusted**

```
\Allfiles> Set-ExecutionPolicy RemoteSigned
```

[!NOTE] If prompted answer with A [Yes to All]

1. Set-Location C:\LabFiles\Allfiles
2. File -> Open **Setup.ps1**
3. Edit Setup.ps1

If you are using an Authorized Lab Host (ALH) lab, you will have a shared licensed production tenant containing Dynamics 365 with pre-created users. Set the following variables in Setup.ps1:

- \$LabAdminPowerLicense="DYN365_ENTERPRISE_PLAN1"
- \$ALH=\$true

If you are using an trial tenant, you will need to have created a Power Platform Per User Plan trial. Set the following variables in Setup.ps1:

- \$LabAdminPowerLicense="POWERAPPS_PER_USER"
- \$ALH=\$false

4. Run **Setup.ps1** by pressing F5 or by clicking the play icon

[!NOTE] If prompted answer with A [Yes to All]

```

.\\Setup.ps1 \\Setup.ps1
### Prepare to run Start-PL600-Setup ##
Start-PL600-Setup -TenantName 'WWLLAB0XX' -CDSLocation 'unitedstates' -UserCount 24
Parameters details for Start-PL600-Setup:
+ TenantName : This is the name portion of name.onmicrosoft.com
+ CDSLocation: This must match be appropriate for Region e.g. US = unitedstates
+ UserCount: This is a number between 1 and 75 that is attending your event
+ APIUrl : You can find the url for your tenant region here if not in US - https://docs.microsoft.com/en-us/dynamics365/customer-engagement/developer/online-management-api/get-started-online-management-api
+ You can find out your tenant region by running running Get-MsolCompanyInformation and looking at CountryLetterCode
### Ready for you to run Start-PL600-Setup ##

```

1. Run the following command:

Start-PL600-Setup -TenantName -CDSLocation -UserCount

Your Tenant name is the name in your tenant credentials. You should enter only the first part of the Tenant name i.e., without onmicrosoft.com e.g., WWLLAB015.

You should configure all 24 users, so set UserCount to 24.

The region should be the region of your tenant. You can find your region by running the commands:

```

Connect-MsolService
Get-MsolCompanyInformation

```

and checking the country code. Valid values for region are:

```

unitedstates
southamerica
canada
europe
asia
australia
japan
india
unitedkingdom
france

```

The ALH labs will be in the unitedstates region

The Setup.ps1 script creates environments and databases for each student and a development environment with the solution imported.

Check Dataverse environments

1. Navigate to the Power Platform Admin Center
<https://admin.microsoft.com/AdminPortal/Home>
2. Verify that there is an environment for each student

Environment	Type	State	Region	Created on	Created by
Prod - veronicaq	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - dani	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - spencerl	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - karenb	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - alans	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - gregw	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - jeffh	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - sanjays	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - kellyk	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - christag	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - mollyc	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - carlosg	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - dianep	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Device Ordering Development	Trial (27 days)	Ready	United States	04/20/2021 8:27 PM	System Administrator
WWLLAB015	Production	Ready	United States	04/16/2021 8:49 AM	System Administrator

[!NOTE] Capacity limits on the tenant may prevent all the environments from being created. If this happens you need to add a Power Apps trial to the tenant and students will have to create a trial environment themselves in the lab.

1. Add a Power Apps trial
 - a. Navigate to Microsoft 365 admin center:
<https://admin.microsoft.com/AdminPortal/Home#>
 - b. Go to **Billing** and then **Purchase Services**
 - c. Search for "**Power Apps**"
 - d. Select **Power Apps per user plan**
 - e. Click **Start free trial**
 - f. Click **Try now**
 - g. Click **Continue**

2. Add a Power Apps license to users

- a. Navigate to Microsoft 365 admin center:
<https://admin.microsoft.com/AdminPortal/Home#>
- b. Go to **Users** and then **Active Users**
- c. Select all users
- d. Select **Manage product licenses**
- e. Select **Assign more**
- f. Select **Power Apps per user plan**
- g. Click **Save changes**

Create a production environment for instructor

1. Navigate to the Power Platform Admin Center:
<https://admin.powerplatform.microsoft.com>
2. Select **Environments**
3. Click on **+ New**

New environment

Name *

Choose name

Type *

Trial (subscription-based) 

Region *

Local environments can provide quicker data access.

United States - Default 

Purpose

Describe the environment purpose

Create a database for this environment?

Yes

Database must always be created for the selected environment type.

[Next](#)

[Cancel](#)

1. In the New environment pane, enter the following details and click **Next**

- a. Name: **Prod - instructor**
- b. Type: **Trial (subscription-based)**

2. Click **Save**

Check development environment

1. Navigate to the Maker Portal <https://make.powerapps.com/>
2. Select the **Device Ordering Development** environment

[!NOTE] If you do not see the environment, press Ctrl-F5 to refresh your browser

1. Click on **Solutions** and verify the Contoso Device Order Management solution has been imported

Display name	Name	Created	Version	Managed externally?
Contoso Device Order Management	ContosoDeviceOrder...	1/1/2023	1.0.0.0	ⓘ
Power Apps Checker Base	msdyn_PowerAppsC...	1/1/2023	12.0.64	ⓘ
Power Apps Checker	msdyn_PowerAppsC...	1/1/2023	12.0.64	ⓘ
Contextual Help Base	msdyn_ContextualH...	1/1/2023	1.0.0.12	ⓘ
Contextual Help	msdyn_ContextualH...	1/1/2023	1.0.0.12	ⓘ
Common Data Services Default Solution	CommonDS	1/1/2023	1.0.0.0	ⓘ
Default Solution	Default	1/1/2023	1.0	ⓘ

1. Click on **Apps** and verify the Device Ordering App and the Device Procurement apps have been imported

Name
Device Ordering App
Device Procurement

Add users to the development environment

1. Navigate to the Power Platform Admin Center <https://aka.ms/ppac>
2. Select the **Device Ordering Development** environment
3. Click on **Settings**
4. Expand **Users + permissions**
5. Select **Users** If the student users are listed, skip to the Assign security roles step.
6. If using an ALH lab, click on **+ Add user**, search and add the following users:
 - a. Alan Steiner
 - b. Alicia Thomber
 - c. Allie Bellew
 - d. Amy Alberts
 - e. Annie Weiler
 - f. Carlos Grilo
 - g. Christa Geller
 - h. Dan Jump
 - i. David So
 - j. Diane Prescott
 - k. Eric Gruber
 - l. Greg Winston
 - m. Jamie Reding
 - n. Jeff Hay
 - o. Julian Isla
 - p. Karen Berg
 - q. Kelly Krout

r. Molly Clark

s. Renee Lo

t. Sanjay Shah

u. Spencer Low

v. Sven Mortensen

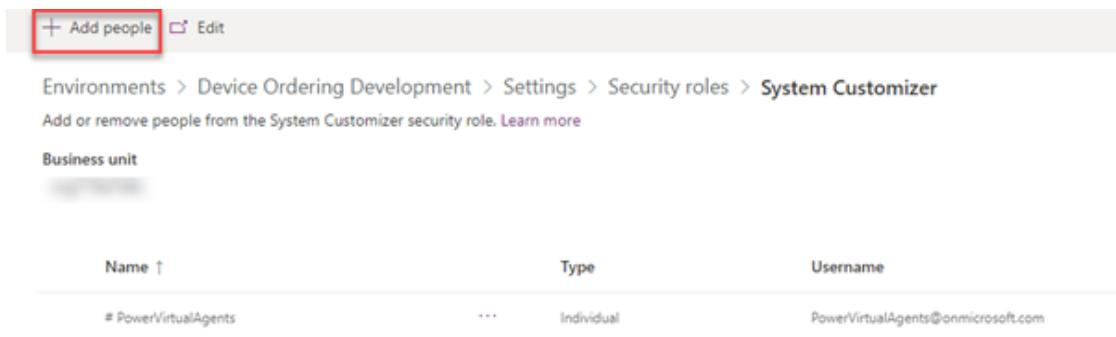
w. Veronica Quek

x. William Contoso

7. If using a trial tenant, click on **+ Add user**, search and add each of the Lab Admin users

Assign security roles

1. Navigate to the Power Platform Admin Center <https://aka.ms/ppac>
2. Select the **Device Ordering Development** environment
3. Click on **Settings**
4. Expand **Users + permissions**
5. Select **Security Roles**
6. Select the **System Customizer** role



The screenshot shows the 'Security roles' page for the 'System Customizer' role. At the top, there are buttons for '+ Add people' (which is highlighted with a red box) and 'Edit'. Below this, the breadcrumb navigation shows: Environments > Device Ordering Development > Settings > Security roles > System Customizer. A note says 'Add or remove people from the System Customizer security role. Learn more'. Under 'Business unit', there is a placeholder text '...' followed by a redacted business unit name. The main table lists one user: # PowerVirtualAgents (Type: Individual, Username: PowerVirtualAgents@onmicrosoft.com). The table has columns: Name ↑, Type, and Username.

Name ↑	Type	Username
# PowerVirtualAgents	Individual	PowerVirtualAgents@onmicrosoft.com

1. If using an ALH lab, click **+ Add people**, search and add the following users:
 - a. Alan Steiner
 - b. Alicia Thomber
 - c. Allie Bellew
 - d. Amy Alberts
 - e. Annie Weiler
 - f. Carlos Grilo
 - g. Christa Geller
 - h. Dan Jump
 - i. David So
 - j. Diane Prescott
 - k. Eric Gruber
 - l. Greg Winston

m. Jamie Reding

n. Jeff Hay

o. Julian Isla

p. Karen Berg

q. Kelly Krout

r. Molly Clark

s. Renee Lo

t. Sanjay Shah

u. Spencer Low

v. Sven Mortensen

w. Veronica Quek

x. William Contoso

2. If using a trial tenant, click on **+ Add people**, search and add each of the Lab Admin users.

3. You are now ready to run the ALM lab **App composition and solution segmentation lab setup**

Lab Test Environment

As this is a shared environment, some tasks that require a tenant Global Administrator or a Service Administrator role will need to be completed by the instructor prior to the course

MVP Solution

1. From Maker portal: <https://make.powerapps.com>
 - a. Select the production environment and go to **Solutions**
 - b. Click **Import**
 - c. Browse to the **SecuritySystem_1_0_0_0.zip** solution file
 - d. Click **Next**
 - e. Click **Import**
 - f. **Publish** the solution

Test data

1. From Maker portal center: <https://make.powerapps.com>
2. Select the production environment and click **Data**
3. Country data
 - a. Select **Country** table
 - b. Select the **Data** tab
 - c. Add 3 records for Tunisia, Morocco, and Algeria
4. Building data
 - a. Select **Building** table
 - b. Select the **Data** tab
 - c. Add one building record for Tunisia
 - d. Add two building records for Morocco
5. Room data
 - a. Select **Room** table
 - b. Select the **Data** tab
 - c. Add two room records for the building in Tunisia
 - d. Add one room record for each of the buildings in Morocco
6. Visit data
 - a. Select **Visit** table
 - b. Select the **Data** tab
 - c. Create 5 visit records for the third day, last day, and the day after the course

Robotic Process Automation lab setup

Lab Test Environment

As this is a shared environment, some tasks that require a tenant Global Administrator or a Service Administrator role will need to be completed by the instructor prior to the course

Power Automate Desktop

1. Power Automate trial
 - a. From Microsoft 365 admin center:
<https://admin.microsoft.com/AdminPortal/Home#>
 - i. Go to **Billing** and then **Purchase Services**
 1. Search for "**Power Automate**"
 2. Select **Power Automate per user plan with attended RPA**
 3. Click **Start free trial**
 4. Click **Try now**
 5. Click **Continue**
 - ii. Go to **Billing** and then **Purchase Services**
 1. Search for **Unattended**
 2. Select **Power Automate unattended RPA add-on trial**
 3. Click **Start free trial**
 4. Click **Try now**
 5. Click **Continue**
 - iii. Go to **Users** and then **Active Users**
 1. Select all users
 2. Select **Manage product licenses**
 3. Select **Assign more**
 4. Select **Power Automate per user plan with attended RPA**
 5. Click **Save changes**
2. Power Automate capacity
 - a. From Power Platform admin center:
<https://admin.powerplatform.microsoft.com>
 - i. Go to **Resources** and then **Capacity**
 1. Click the **Add-ons** tab
 2. Click **+ Assign to an environment**
 3. Select the production environment
 4. Assign **25** to **Power Automate Unattended RPA**
 5. Click **Save**

Security roles

1. Assign security roles to users
 - a. From Power Platform admin center:
<https://admin.powerplatform.microsoft.com>
 - i. Select the production environment and go to **Settings**
 1. Expand **Users + permissions**
 2. Click **Security roles**
 3. Select **Sales Enterprise app access** role
There should be 25 users listed. If not click **+ Add people**, search and add the users (see the list below)
 4. Click **Security roles**
 5. Select **Salesperson** role
There should be 25 users listed. If not click **+ Add people**, search and add the following users:
- a. Alan Steiner
- b. Alicia Thomber
- c. Allie Bellew
- d. Amy Alberts
- e. Annie Weiler
- f. Carlos Grilo
- g. Christa Geller
- h. Dan Jump
- i. David So
- j. Diane Prescott
- k. Eric Gruber
- l. Greg Winston
- m. Jamie Reding
- n. Jeff Hay
- o. Julian Isla
- p. Karen Berg
- q. Kelly Krout
- r. Molly Clark
- s. Renee Lo
- t. Sanjay Shah
- u. Spencer Low
- v. Sven Mortensen
- w. Veronica Quek
- x. William Contoso

Sample data

1. Sample Data
 - a. From Power Platform admin center:
<https://admin.powerplatform.microsoft.com>
 - i. Select the production environment and go to **Settings**
 1. Expand **Data Management**
 2. Click **Sample data**
 3. Click **Install Sample Data**
 4. Click **Close**
2. Products
 - a. From Power Platform admin center:
<https://admin.powerplatform.microsoft.com>
 - i. Select the production environment and click **Open environment**
 1. Select **Sales Hub**
 2. Navigate to **Products**
 3. Select the **Draft Products, Families, and Bundles** view
 4. Select the **Office 365 Service (sample)** product
 5. Click **Publish** and **Publish hierarchy**
 6. Click **Confirm**
3. Price List
 - a. From Power Platform admin center:
<https://admin.powerplatform.microsoft.com>
 - i. Select the production environment and click **Open environment**
 1. Select **Sales Hub**
 2. Select **App Settings** area
 3. Navigate to **Price Lists**
 4. Select the **CRM Service USA (sample)** price list
 5. Select the **Price List Items** tab
 6. Click **+ Add New Price List Item**
 7. Select the **Exchange Online (sample)** product
 8. Select the **Pricing information** tab
 9. Enter **9** in the **Amount** field
 10. Click **Save and Close**
 11. Click **+ Add New Price List Item**
 12. Select the **SharePoint Online (sample)** product
 13. Select the **Pricing information** tab
 14. Enter **12** in the **Amount** field
 15. Click **Save and Close**
 16. Click **Save and Close**

4. Invoices

a. From Power Platform admin center:

<https://admin.powerplatform.microsoft.com>

i. Select the production environment and click **Open environment**

1. Select **Sales Hub**

2. Navigate to **Invoices**

3. Click **+ New** and enter the following details

a. Name: **Exchange Licenses**

b. Customer: **Alpine Ski House (sample)**

c. Currency: **US Dollar**

d. Price List: **CRM Service USA (sample)**

e. Click **Save**

f. Click **+ Add Product**

g. Select the **Exchange Online (sample)** product

h. Enter **100** in the **Quantity** field

i. Click **Save and Close**

j. Click **Save and Close**

4. Click **+ New** and enter the following details

a. Name: **SharePoint Licenses**

b. Customer: **Coho Winery (sample)**

c. Currency: **US Dollar**

d. Price List: **CRM Service USA (sample)**

e. Click **Save**

f. Click **+ Add Product**

g. Select the **SharePoint Online (sample)** product

h. Enter **250** in the **Quantity** field

i. Click **Save and Close**

j. Click **Save and Close**

5. Click **+ New** and enter the following details

a. Name: **Office 365 Licenses**

b. Customer: **Fourth Coffee (sample)**

c. Currency: **US Dollar**

d. Price List: **CRM Service USA (sample)**

e. Click **Save**

f. Click **+ Add Product**

g. Select the **Exchange Online (sample)** product

h. Enter **50** in the **Quantity** field

i. Click **Save and Close**

j. Click **+ Add Product**

k. Select the **SharePoint Online (sample)** product

l. Enter **50** in the **Quantity** field

m. Click **Save and Close**

n. Click **Save and Close**

Instructions for creating a Trial tenant

The labs require a Microsoft 365 tenant with:

- **Power Platform:** A Software-as-a-Service (SaaS) application platform that enables power users in line of business roles to easily build and deploy custom business apps.
- **Microsoft Dataverse:** Make it easier to bring your data together and quickly create powerful apps using a compliant and scalable data service and app platform that integrates into Power Apps.

[!IMPORTANT] If you are using a hosted lab provided by an Authorized Lab Host (ALH) then you do not need to perform these tasks and create a trial.

Location of files

Files for the course are in GitHub <https://github.com/MicrosoftLearning/PL-600-Power-Platform-Solution-Architect/tree/master/Allfiles>

Credentials

[!IMPORTANT] **IMPORTANT** Do not use your own company credentials or company name when creating a trial. We recommend using a Gmail or Outlook address rather than your work email address to prevent your trial being associated with your company.

Profiles or InPrivate/Incognito Mode

One of the issues faced with trials is credential leakage where you suddenly find yourself in your own company's Microsoft 365 live tenant and environment.

You have two options:

- Use InPrivate (Edge) or Incognito (Chrome)
- Create a Profile (Edge) or add a Person (Chrome)

[!NOTE] InPrivate/Incognito does not store session cookies, and this can cause authentication issues with constant login prompts.

How to create a person <https://support.google.com/chrome/answer/2364824>

How to create a profile <https://www.onmsft.com/how-to/how-to-use-profiles-a-new-feature-in-microsoft-edge-insider> or
<https://www.tenforums.com/tutorials/144642-how-add-profile-microsoft-edge-chromium.html#option1>

Trial tenant

Sign up for a trial

1. In a browser profile session, navigate to
<https://signup.microsoft.com/SignedUp?OfferId=83D3609A-14C1-4FC2-A18E-0F5CA7047E46>

Thank you for choosing **Power Apps per user plan**

The screenshot shows the first step of a four-step sign-up process. Step 1, 'Let's set up your account', is highlighted with a red circle around the number 1. It asks for a work or school email address to check if a new account is needed. A large input field is provided for the email, and a 'Next' button is below it. To the left of the input field, there is a vertical dotted line connecting the steps. To the right of the input field, the remaining three steps are listed: 'Tell us about yourself', 'Create your business identity', and 'You're all set'.

① Let's set up your account

Enter your work or school email address, we'll check if you need to create a new account for Power Apps per user plan.

Enter your email address

Next

② Tell us about yourself

③ Create your business identity

④ You're all set

For this screen, use a valid email address (**not your business email address**) where you can receive credentials and password resets, and a valid mobile phone number. If you are already using Microsoft 365 then use another email address e.g., outlook.com or gmail.com.

1. Click **Next** and then click **Setup account**

(2)

Tell us about yourself



First name

Middle name

Last name

Business phone number

Company name

Power Platform Training

Company size

1 person

Country or region

Next

1. Fill out the form as follows:

2. Use **Power Platform Training** for **Company name**

3. Select **1 person** for **Your organization size**

[!NOTE] We recommend using United States for country and English as language as this makes following the labs easier.

1. Click **Next**

2. Select the **Text me** radio button, enter your mobile number, and click **Send Verification Code**

3. Enter the unique verification code you received and click **Verify**

(3)

Create your business identity



To set up your account, you'll need a domain name. [What is a domain?](#)

You'll probably want a custom domain name for your business at some point. For now, choose a name for your domain using [onmicrosoft.com](#)

yourbusiness

.onmicrosoft.com

Check availability

Next

1. Enter a unique name to reflect this training e.g., **PL600 + your initials + year/month for your business**

2. Click **Check availability**

3. Click **Next**

(3) Create your business identity

Now create your user ID and password to sign in to your account. ⓘ

Name: [REDACTED]@onmicrosoft.com

Create password:

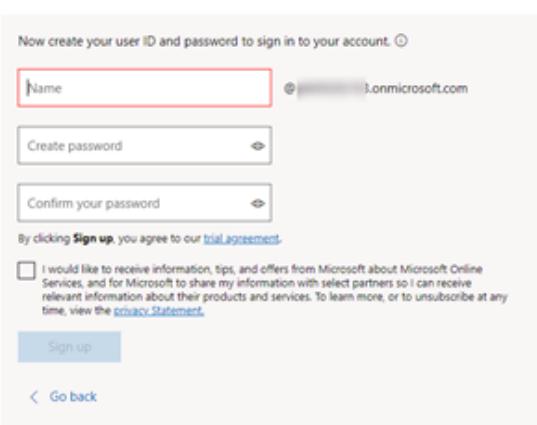
Confirm your password:

By clicking **Sign up**, you agree to our [trial agreement](#).

I would like to receive information, tips, and offers from Microsoft about Microsoft Online Services, and for Microsoft to share my information with select partners so I can receive relevant information about their products and services. To learn more, or to unsubscribe at any time, view the [privacy statement](#).

Sign up

< Go back



1. Fill out the form

2. Click **Sign up**

Thank you for choosing **Power Apps per user plan**

- (1) Signup started
- (2) Nice to meet you, [REDACTED]
- (3) Thanks for creating an account with us, [REDACTED]
- (4) You're all set

Thanks for signing up for a Power Apps per user plan trial.

Your username is [REDACTED]@[REDACTED].onmicrosoft.com.

We've sent a confirmation with your sign-in information to [REDACTED]

Get Started

[Manage your subscription](#)

1. Write the user ID down

2. Click **Get Started**

Configure Microsoft 365

1. Navigate to the Microsoft 365 Admin Centre

<https://admin.microsoft.com/AdminPortal/Home>

1. Click Users

2. Click Active Users

Verify that your user has all licenses assigned

1. Close the pane

You are now ready to perform the ALM setup instructions.

Rename default environment

1. Navigate to the Power Platform admin centre
<https://admin.powerplatform.microsoft.com>
2. Select Environments
3. Click on the default environment which will be named after the name you entered for your business

4. Click **Edit**
5. Change the environment name to **Personal Productivity**
6. Click **Save**

Exercise Overview

Split into groups of 3-5 students. Select a company to research. Choose a large, publicly traded, company. The groups will then share their findings and discuss the process of researching. Assume you need this information to bring with you to a meeting. The meeting could be the first meeting with the potential customer; it could be a meeting with your peers to discuss a potential deal.

Learning Objectives

After completing the exercise, you will be able to:

- Find information about potential customers
- Evaluate the value of this information
- Understand how others might complete the same tasks

Exercise 1: Learn about the customer

Find the following information regarding your selected company. Make sure to take notes so you can share what you've found with the larger group. It is not necessary to gather ALL of the items listed here. This is an example list of key details that can help you better know your customers.

People

Identify three separate c-level executives.

1. Name
2. Contact info and/or social media
3. Prior work
4. (Bonus) School background

Corporate

Identify stock symbol and value

Locate earnings report/call

Identify key divisions or lines of business

Identify headquarters and office locations

Culture/News

Read recent company press release

Locate social media accounts-official and unofficial

Locate what others say about company, news and social. Identify top pain points.

Technology

Do they have any preferred cloud providers?

Any publicly stated technology preferences?

Exercise Overview

During this exercise, you will be reviewing project details for Contoso. This includes a set of requirements gathered from customer workshop sessions. You will use this information to design a high-level architecture and complete a fit gap analysis.

Learning Objectives

After completing the exercise, you will be able to:

- Perform evaluation of requirements to decide which approach to take
- Develop a high-level architecture for a set of requirements
- Perform a gap analysis from the Dynamics 365 apps used and the requirements

Exercise 1: Evaluate Project Details

Open and review the Project Details.docx file.

This contains the following information:

- Project Scenario and Overview
- Key Facts
- Requirements

As you review the document your team should take notes as you go and start to develop a proposed design to meet the requirements.

Exercise 2: Complete the Project Workbook

Open and complete the worksheets in the Project Workbook.xlsx file.

Complete each of the following tabs:

- **High-level architecture diagram** Create a simple high-level architecture diagram. This can be a picture of a whiteboard, a photo of a paper drawing. Simply paste it into the worksheet tab.
- **Dynamics 365 apps used** Document which apps you used and a brief description how it will be used.
- **Fit Gap** For each requirement complete the priority, level of effort, category and implementation details.

Your work will be reviewed by another team in a future exercise, so provide the level of detail you would want from the team you review.

Scenario

Contoso Services provides after-market support for various manufacturer's smart beds. They use a proprietary solution today to manage their customers and their support ticket requests. They would like to move to a solution built on Microsoft Power Platform.

Some of their support requests can be handled quickly with a phone call or with a quick email response of a knowledge base article. Some may go on for days (or weeks) as the technicians dig in and research, then solve, the customers' problems. For complex challenges they dispatch technicians onsite to resolve the problem.

Today most of their support is reactive to a customer calling in. They would like to be able to become more proactive.

There are three type of cases that can come into Contoso's service center.

1. Items from customers without support agreements.
2. Items from preferred level support customers.
3. Items from premier level support customers.

Request for support can come from several sources: email, phone, online chat, social media, and so on. Requests for support coming in need to be evaluated and routed accordingly based on the requirements and policies. Some of the manufacturers have apps for customers that communicate with the beds and can then communicate the need for support to Contoso.

Contoso receives a data feed nightly from the manufacturers they support with customer information including bed serial number, bed configuration and ongoing telemetry of bed hardware wear indicators.

Key Facts

FACT	Deteails
Number of solution users	2,500
Number of service center agents	500
Number of field technicians	100 employees, 500 contractors
Locations	United States, United Kingdom and India
Number of customers	1.5 million
Contoso founded	2002
Contoso board of directors	9 members
Devices	Some users work with a single monitor and 1024x768 screen resolution

Requirements

ID	Description
1	As the system, a help request shall be routed to the named team accordingly based on several factors including their service level, and type of incident so that the correct group can address the issue.
2	As a support staff, I should be able to see a list of open help requests so I can pick one to work on for which I am qualified.
3	As a customer, I don't want to wait on hold so should I be able to open a new request from Contoso's website using an interactive chat.
4	As a support staff, I need to be able to search knowledge base articles so that I can send them to customers.
5	As the system, I should process the nightly data feed from the manufacturers and keep track of the customer and bed configuration so requests can properly be associated.
6	As a support staff, I should be able to schedule a technician for onsite service.

ID	Description
7	As a support staff, I should see a list of customers to contact to schedule proactive maintenance based on insights from the bed wear indicators sent to us by the manufacturer so I can provide proactive service to ensure happy customers.
8	As a field technician, I should be able to view my next scheduled customer visit and receive details of the problem and directions if needed so that I can resolve their bed problem.
9	As a support staff, I need to be able to sell a support agreement to customers that didn't buy one with their original bed purchase so I can get their support request assigned to be resolved.
10	As a customer, I should receive an email when a support staff has closed my request so I can provide feedback on the experience.
11	As a support manager, I need to be able to see any support requests that are outside of the SLA for the customer so I can help prioritize the requests.
12	As a customer, I should be able to see the status of my support requests on Contoso's website so that I know what is happening to resolve my problem.
13	As a support manager, I want to receive a copy of any customer communication that the customer seems upset so that I can get involved to help resolve the problem.
14	As a support staff, I need to be able to see the remaining time I have to meet the SLA so that I'm not in trouble with my manager.
15	As the system, I need to track when a support request was opened, number of days open and resolved.
16	As a customer, I need to be able to take ownership of a bed, and support agreement when I acquire a used bed from a private buyer so that I can get the benefits of ownership.
17	As a support staff, regardless of my location, I should have acceptable response time for page loads.
18	As a C-level executive, I need to see real time metrics on the performance of the support team so I know what's going on.

#

Exercise Overview

During this exercise, you will be reviewing the project workbook for the Contoso project created by another team of Solution Architects. Part of this

exercise is deciding how best to manage your time. Make sure you find out from your instructor the time allotted.

Learning Objectives

After completing the exercise, you will be able to:

- Perform evaluation of work produced by others effectively.
- Deliver constructive feedback to others.

Exercise 1: Review Project Workbook Details and Prepare Feedback

You will be assigned another team of Solution Architects to work with and exchange feedback.

Exchange a digital copy of your project workbook from Module 2 where you architected a high-level solution for Contoso with the other team.

As a team, review the other team’s workbook details and develop notes so you can go over the feedback with the other team later. Your notes should consider the following:

- High-level architecture diagram
 - Does it accurately represent the requirements?
 - How could it be improved?
- Dynamics 365 apps used
 - Did they use the appropriate Dynamics 365 apps for the requirements?
 - If you disagree with a decision, why?
- Fit Gap
 - Review the other team’s choices, do any seem wrong?
- Overall assessment
 - What did the team do well?

- What could the team have improved?

Exercise 2: Provide feedback to the other team

Take turns with the other team providing feedback to each other based on your review.

As the team providing feedback:

- Keep the feedback constructive.
- Provide examples of how to improve the proposed architecture
don't just say it's wrong
- Avoid coming across like you know everything!

As the team receiving feedback:

- Listen to the feedback, try not to interrupt.
- Try to not be defensive and listen to the ideas.
- Ask clarifying questions where the feedback isn't clear.

If you have time, identify one requirement that you think is a very poorly written requirement.

Exercise Overview

In this exercise you will work as small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

Learning Objectives

After completing the exercise, you will:

- Evaluate customer requirements
- Make decisions for the data model

Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

- Visitors must have a reserved spot to gain access.
- Visitors invited by sales staff must be tracked to a sales process, visitors just touring for fun are not tracked to a sales process.
- Visitors can bring guests.
- Each visitor must have a photo taken upon arrival and associated with their visit
- Each visitor must sign a waiver of liability each time they visit, and you must store their signature and date time of acceptance.
- Each primary visitor is assigned an engagement tracking device that tracks their location in the manufacturing area and showroom. This is for safety as well as for improving the tours. The device stores the data in its own cloud service that offers both bulk export and API access.
- You must allow for the tracking data to be viewed in the sales process user interface used by the sales staff

- Marketing has asked to be able to view visitors by day/month/quarter along with statistics on closing of sales after a visit.

Tasks

Create a data model for the above requirements. Use whatever tool or medium that you have available to complete the data model. You can use a whiteboard, Visio or pen and paper.

- How did you handle storing of visitor photos?
- How did you handle storing of visitor waiver acceptance and signature?
- How did you accommodate for viewing of tracking data in the sales process?
- Did you use anything from the Common Data Model schema?
- How did you handle marketing's statistics needs?

Exercise Overview

In this exercise you will work in small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

Learning Objectives

After completing the exercise, you will:

- Identify and group app users
- Discuss and propose app types to solve a requirement
- Identify and recommend use of components

Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

Review the requirements here and address the concerns listed later in this document.

- Fabrikam has been rapidly moving to the cloud and already leverages Office 365 for their email. They are adopting the use of Microsoft Teams at a rapid pace.
- The sales staff handles their own leads and opportunities. They are compensated by commission on close of sales and are very competitive. The staff that works on small deals all work together and receive commission on all closed deals in their department. The large deals sales staff mostly work by themselves; except on very large deals where there is a team of them assigned.
- The sales staff that works on large deals regularly visits prospects offices and are having meetings at conferences with prospects. While they each have mobile hotspots for WIFI, in practice many buildings have little or no signal.

- To handle the sales staff viewing tour data a custom connector will be created to connect to the device service cloud API to make the data available to the app the sales staff is using.
- Reception staff must be able to check in visitors prior to their showroom and manufacturing floor tour. During the check in, they must be able to capture a picture of each visitor. Reception staff must not have access to the sales data.
- Discounts can only be approved and applied by one of the sales managers. The app should ensure the sales staff can't approve their own discounts. In the data model, there is a discount field of type currency and a required lookup field to the approver.
- Customers should be able to request a showroom visit from Fabrikam's website. The proposed solution should remind them via SMS message if they provide a phone number 24 hours before their visit.

Discuss as a group how you would address each of the following:

1. Identify the following:
 1. Different app user groupings?
 2. What Power App(s) would you propose creating for these groups?
 3. What type would they be?
 1. model-driven
 2. canvas
 3. portal
 4. non-Power App Custom Dev
2. How are you accommodating in your app architecture for the sales staff that visit prospects' offices?
3. How will you propose to use the custom connector so the sales staff can view the tour data from their app?
4. Identify places a Power Apps Component Framework component might be helpful or required by your application architecture?

5. If any of your apps are canvas apps, what is your plan to support multiple app makers being able to work on building the app?
6. What would you offload and rely on Power Automate to handle?

Exercise Overview

You will be working in small groups. Discuss each scenario and address the question. Each group takes a turn leading the larger group discussion on the proposed approaches.

During this exercise, you will be reviewing some scenarios and determining how to architect them using Power Automate.

Learning Objectives

After completing the exercise, you will be able to:

- Evaluate requirements and propose a Power Automate flow design
- Evaluate a proposed design that includes Power Automate

Scenario 1

When working on a help request for a smart bed customer, Contoso’s support staff can request a replacement bed from the manufacturer. Each manufacturer provides a contact that can authorize the replacement. Once authorized, the support staff can inform the customer.

What would your design look like to implement this process?

Scenario 2

When a repair is required, the system needs to look up the customer’s support plan, check if they have available repair allowance / warranty, and if they do decrease the cost of the repair from their lifetime warranty allowance and prioritize the help request.

Your development team has proposed a plug-in is that the best choice or how might you handle this in Power Automate?

Scenario 3

When a manufacturer has a defect, they issue a recall to fix the problem. They will provide Contoso a file that contains the serial numbers of the affected beds. Each serial number must be looked up to see if there is an active customer and the customer record must be tagged with the recall. For some large recalls this could exceed 200,000+ beds. The process must be designed so it can be restarted if the process fails without creating any duplication.

What would your design look like to implement this process?

Scenario 4

On create of a help request Contoso needs to calculate the customer deductible which is based a percentage determined by how old their bed is. The support staff needs this value displayed as soon as they create the record.

What would your design look like to implement this?

Exercise Overview

In this exercise you will work in small groups. Review the information presented about Fabrikam Robotics and complete the tasks.

Learning objectives

After completing the exercise, you will:

Exercise 1

You are building a solution for Fabrikam to track visitors to a showroom and manufacturing site. Some of the visitors are potential purchasers and some are just there to see the magic of the robots working.

Review the requirements here and address the concerns listed later in this document.

- Fabrikam has been rapidly moving to the cloud and already leverages Office 365 for their email. They are adopting the use of Microsoft Teams at a rapid pace.
- You have already architected the solution to include the following Power Apps:
 - A model-driven app named Sales Central to be used by the sales staff and sales managers.
 - A canvas app to check in guests visiting the showroom to be used by the reception staff.
 - A portal that visitors can use to request to visit.
- Fabrikam is going to use Power Apps per app licenses for reception staff and Dynamics 365 Sales app licenses for the sales staff.
- Fabrikam’s IT security team wants to make sure the new solution doesn’t open up any security exposure from potential data leaks. You must ensure that your security setup prevents users from building their own flows in Power Automate that send data to outside services that

aren't approved. You could almost see the panic on their faces when your sales team did the demo of how easy it was to build a flow using random 3rd party connectors.

- The sales staff handles their own leads and opportunities. They are compensated by commission on close of sales and are very competitive. The staff that works on small deals all work together and receive commission on all closed deals in their department. The large deals sales staff mostly work by themselves; except on very large deals where there is a team of them assigned.
- Reception staff must not have access to the sales data.
- Sales managers need to have access to the deals of their team members.
- Discounts can only be approved and applied by one of the sales managers. The app should ensure the sales staff can't approve their own discounts. In the data model, there is a discount column of type currency and a required lookup column to the approver.
- Fabrikam's organization has the following divisions and departments:
 - Sales
 - Small Deals
 - Large Deals
 - Wholesale
 - Marketing
 - Operations
 - Manufacturing
 - Showroom (has all the reception staff)
 - Shipping
 - Customer Service

Discuss as a group how you would address each of the following:

1. How do you calm the concerns of the security staff and the rogue connector use?
2. How do you handle ensuring only sales managers approve and apply discounts?
3. How would you control access to each of the Power Apps?
4. What do you need to do to accommodate that Power Apps per app licenses are used for the reception staff?
5. What CDS Security Roles would you have the team create?
6. What business units can you begin to define to support your solution?
7. How will you handle so portal users can only see their own visit request records to check the status and potentially cancel the visit?
8. How would you ensure the small sales department can see all the data for all small sales staff, but the large deals sales staff can only see their own except when a team is put together to handle very large deals?

Exercise

Evaluate requirements for first-party apps

Review the requirements for each app. Decide how best to address a requirement. Some items might be out of the box functionality, some might have nothing to do with our first-party applications, some might need third-party apps, and more.

Dynamics 365 Sales

- While viewing a contact record, sales managers need to see detailed information about the parent account record of that contact.
- Preferred pricing is only valid for a period of time and reverts back upon agreement expiration.
- Sales reps need to have the ability to configure products (color, size, components, etc.) and track them to an opportunity. We currently use a custom-built product configurator. It was built by a developer that is still on staff. It's a WPF application built using .NET.
- Some users expect the opportunity form to show and hide columns based on their column choices on the opportunity form.
- Display real-time information of an account's top current events on the account form.
- When a user creates a new opportunity valued at greater than \$1 million, change the owner of the record to the user's team. Notify the original record owner of the change.
- You must display the current stage of the lead in the lead grid.

Dynamics 365 Customer Service

- When a customer calls in for a new support request, if they do not have available pre-purchased support tickets available, support agents need to be able to sell them support tickets without leaving the case record they are currently viewing.
- When a preferred customer is out of available support incidences we need to reach out and offer to sell them more.
- Users must not access the application on mobile devices.
- Once a support request is opened for an on-site service, we need to dispatch to an available qualified field resource.
- The top support agent (the one with the highest number of closed case records) has created a personal chart to show their open cases from the beginning of the week. You need to make this available to all support agents to see their own support cases from the beginning of the week.
- You must display the current stage of the case in the case grid.
- Some users expect the case form to show and hide sections based on their column choices on the opportunity form.

Dynamics 365 Field Service

- We have implemented a new training program and need resources' skills automatically updated when they pass a class.
- All screens must render for the user in less than 5.25 seconds.
- Resources must be certified prior to completing field tickets without a supervisor.
- Each work order requires a work order type.
- Dispatchers perform 70% of the booking, technicians can perform self-booking via mobile.
- Many customers do not accurately represent what is actually the problem that requires service. We need to make better use of our technicians' time and only send them on a call when it's needed.

Exercise

Evaluate requirements for chatbots

Review the requirements for each scenario. Decide how best to address the requirement with chatbots.

Scenario: Product and pricing

The organization has a complex set of products and prices. Sales users find it difficult to find information about which products can be sold together and how to determine the correct pricing.

You have the following set of requirements:

- Sales people must be able to find details about products easily.
- Sales people need to be shown the pricing rules for each product.
- The rules for products and prices vary by country.

What should you use to meet these requirements?

Scenario: Customer service

The customer service department is overwhelmed with calls. You have been tasked with improving productivity of the department and increasing customer satisfaction.

You have the following current systems:

- Dynamics 365 Customer Service is installed for basic ticket management.
- Legacy website that allows customers to login and see their tickets and submit new tickets. New tickets are created in Dynamics 365 but there is no further integration.
- SharePoint library with support processes and other documentation.

You have the following set of requirements:

- The website only shows the basic ticket details that were submitted. Customers must be able to easily obtain the status and latest action for their tickets.
- Customers have requested a search capability for support information on the website.
- Easily answered questions should be handled by a bot.
- Customers should be able to be transferred to a human agent with the bot conversation available to the human agent.

What should you use to meet these requirements?

lab: title: 'Lab 00: Validate lab environment'

Lab 0 - Setup for ALM lab

1. Get a new Azure Pass (valid for 30-days) from the instructor or other source.
2. Use a private browser session, go to Microsoftazurepass.com to redeem your Azure Pass using the tenant credentials provided to you). (Redeem a Microsoft Azure Pass) Follow the instructions for redemption.
3. Using the same browser session, go to portal.azure.com, then search for **Azure DevOps**. In the resulting page, click Azure DevOps Organizations.
4. Next, click on the link called **My Azure DevOps Organizations** (or navigate to <https://aex.dev.azure.com/>).
5. Create a new organization (find blue box in upper right-hand corner of the screen) using the Default directory.
6. Choose the newly created organization, then choose Organization settings on the left-hand side of the screen
7. Navigate to Organization settings -> Billing -> Setup billing -> Select an Azure subscription, then select the Azure Pass subscription, then choose **MS Hosted CI/CD** and set the field **Paid parallel jobs** to 1. Then click SAVE in the blue box at the bottom.
8. Wait at least 3 hours before using the CI/CD capabilities so that new settings are reflected in the back end. Otherwise you will still see the message **This agent is not running because you have reached the maximum number of requests**.
9. As an optional step, you can validate this by creating a new pre-defined project using the newly created org with billing enabled, using <https://azureddevopsdemogenerator.azurewebsites.net/>. Wait for some time before trying, then run a test build. [Not necessary for PL-600]

lab: title: 'Lab 01: Application Lifecycle Management'

Module 8: Application Lifecycle Management

Scenario

In this hands-on lab, you are a solutions architect for Contoso, helping them adopt the Power Platform.

The team building the "Device Order Management" app is now ready to transport the solution from their development environment to the test environment for testing.

In this lab, you will be using Azure DevOps and the Power Platform build tools to automate checking the solution into a source control repository and then use that to deploy to test and production environment.

Lab Test Environment

You will be assigned one or more users to use to complete the tasks. As this is a shared environment, some tasks that require a tenant Global Administrator or a Service Administrator role will have already been performed.

High-level lab steps

As part of configuring Azure DevOps ALM automation, you will complete the following

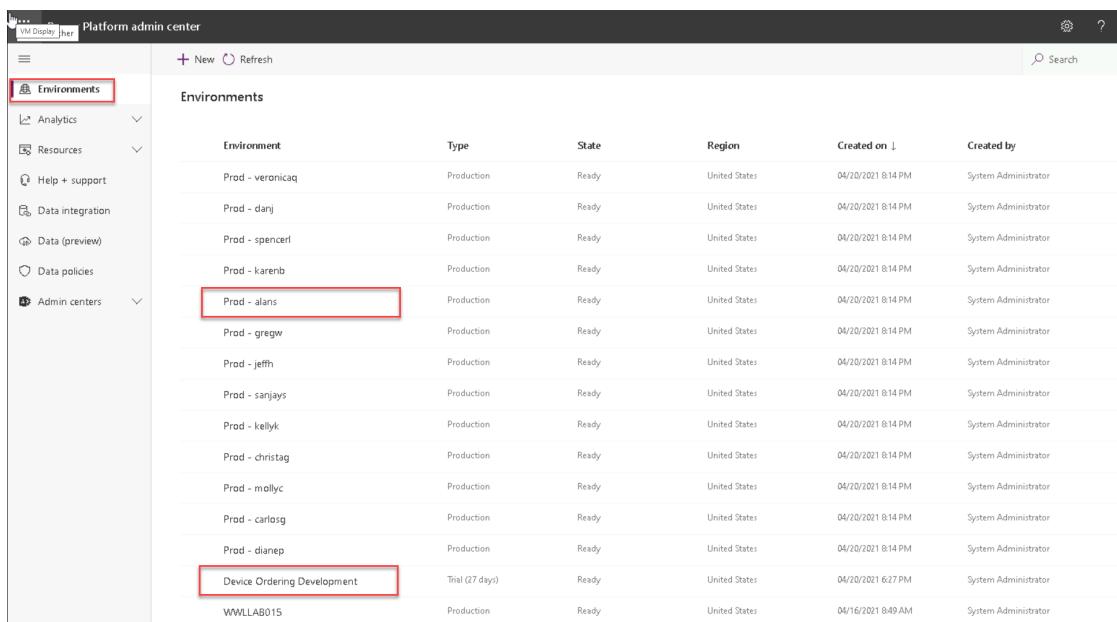
- Sign up for an Azure DevOps account
- Create an Azure DevOps project
- Configure Power Platform ALM tasks
- Build an export solution pipeline
- Test the export from dev to Azure DevOps
- Build a release pipeline

Exercise 1: Environments

In this exercise, you verify that you have access to the DEV environment and verify that you have a PROD environment.

Task 1: Verify environments

1. Navigate to the Power Platform Admin Center:
<https://admin.powerplatform.microsoft.com>
2. Provide your tenant credentials and click **Next**
3. Provide your password and click **Sign in**
4. Click **Yes**
5. Select **Environments**



Environment	Type	State	Region	Created on	Created by
Prod - veronicaq	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - danj	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - spencerl	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - karenb	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - alans	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - gregw	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - jeffh	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - sanjays	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - kellyk	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - christag	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - mollyc	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - carlog	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Prod - dianep	Production	Ready	United States	04/20/2021 8:14 PM	System Administrator
Device Ordering Development	Trial (67 days)	Ready	United States	04/20/2021 8:14 PM	System Administrator
WWLLAB015	Production	Ready	United States	04/16/2021 8:49 AM	System Administrator

[!NOTE] You should see the Development environment which is called **Device Ordering Development**. If you cannot see this environment, contact your instructor.

You should see a Production environment named after your tenant user. For example, if your user is Alan Steiner, there should be an environment called **Prod >- alans**.

1. If you do not have a PROD environment, click on **+ New**

New environment

Name *

Type *



Region *

Local environments can provide quicker data access.



Purpose



Create a database for this environment?



Yes

Database must always be created for the selected environment type.

[Next](#)

[Cancel](#)

1. In the **New environments** pane, enter the following details and click **Next**

- a. Name: **Your user's name prefixed with Prod** e.g., "Prod - alans"
 - b. Type: **Trial (subscription-based)**
 - c. Create a database: **On**
2. Click **Save**

Task 2: Environment URLs

1. Navigate to the Power Platform Admin Center:
<https://admin.powerplatform.microsoft.com>
2. Select **Environments**
3. Select the **Device Ordering Development** environment
4. Copy the **Environment URL** into a notepad
5. Select **Environments**
6. Select your **Prod** environment
7. Copy the **Environment URL** into a notepad

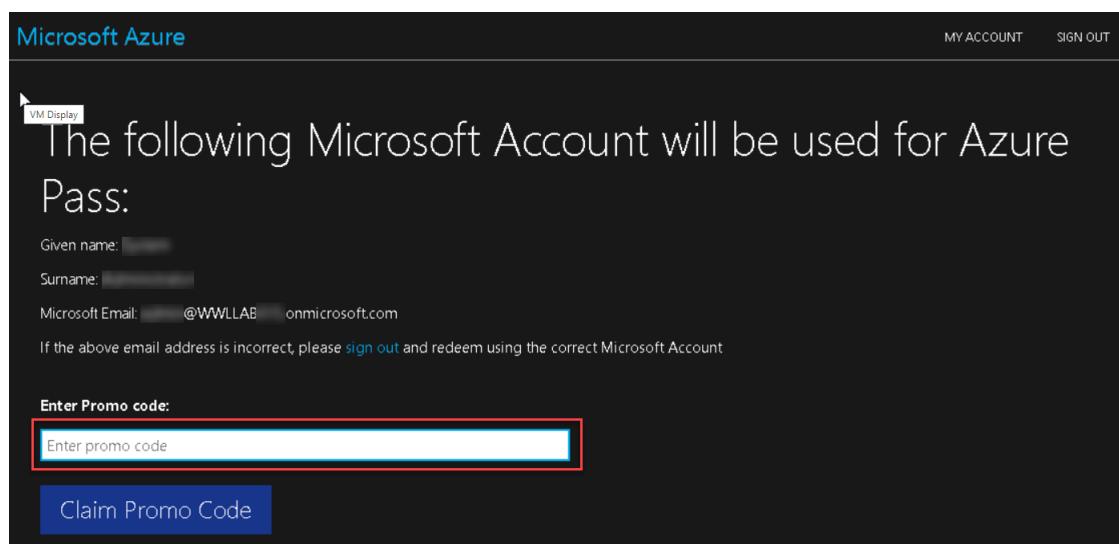
Exercise 2: Azure Subscription

In this exercise, you will sign up for an Azure subscription using an Azure 30-day Pass.

[!NOTE] An Azure Pass is required to provide Azure credits for use with Azure DevOps pipelines

Task 1: Redeem Azure Pass

1. Get an Azure Pass from your learning provider/instructor.
2. Navigate to <https://www.microsoftazurepass.com>
3. Click **SIGN IN**
4. Provide your tenant credentials and click **Next**
5. Provide your password and click **Sign in**
6. Click **Yes**
7. Click **CONFIRM MICROSOFT ACCOUNT**



1. Enter the Azure Pass code and click **Claim Promo Code**
2. Follow the steps at <https://www.microsoftazurepass.com/Home/HowTo> to redeem the Azure Pass

Exercise 3: Initialize Azure DevOps

In this exercise, you will setup an Azure DevOps account and configuring the Power Platform build tools for the account.

[!NOTE] If you already have an Azure DevOps organization outside of this course and this environment, you **CANNOT** use that organization for this lab. You will need to follow the below instructions to sign up.

Task 1: Setup Azure DevOps

1. Sign up for Azure DevOps
 - a. Navigate to Azure Devops <https://dev.azure.com>
 - b. Click **Sign in**
 - c. Provide your admin credentials and sign in.

Azure DevOps

Plan smarter, collaborate better, and ship faster with a set of modern dev services.

[Start free >](#)

[Start free with GitHub >](#)

Already have an account?

[Sign in to Azure DevOps >](#)

- d. Click **Sign into Azure DevOps**

We need a few more details

Your name:

Alan Steiner

We'll reach you at:

alans|WWLLAB015.onmicrosoft.com

From:

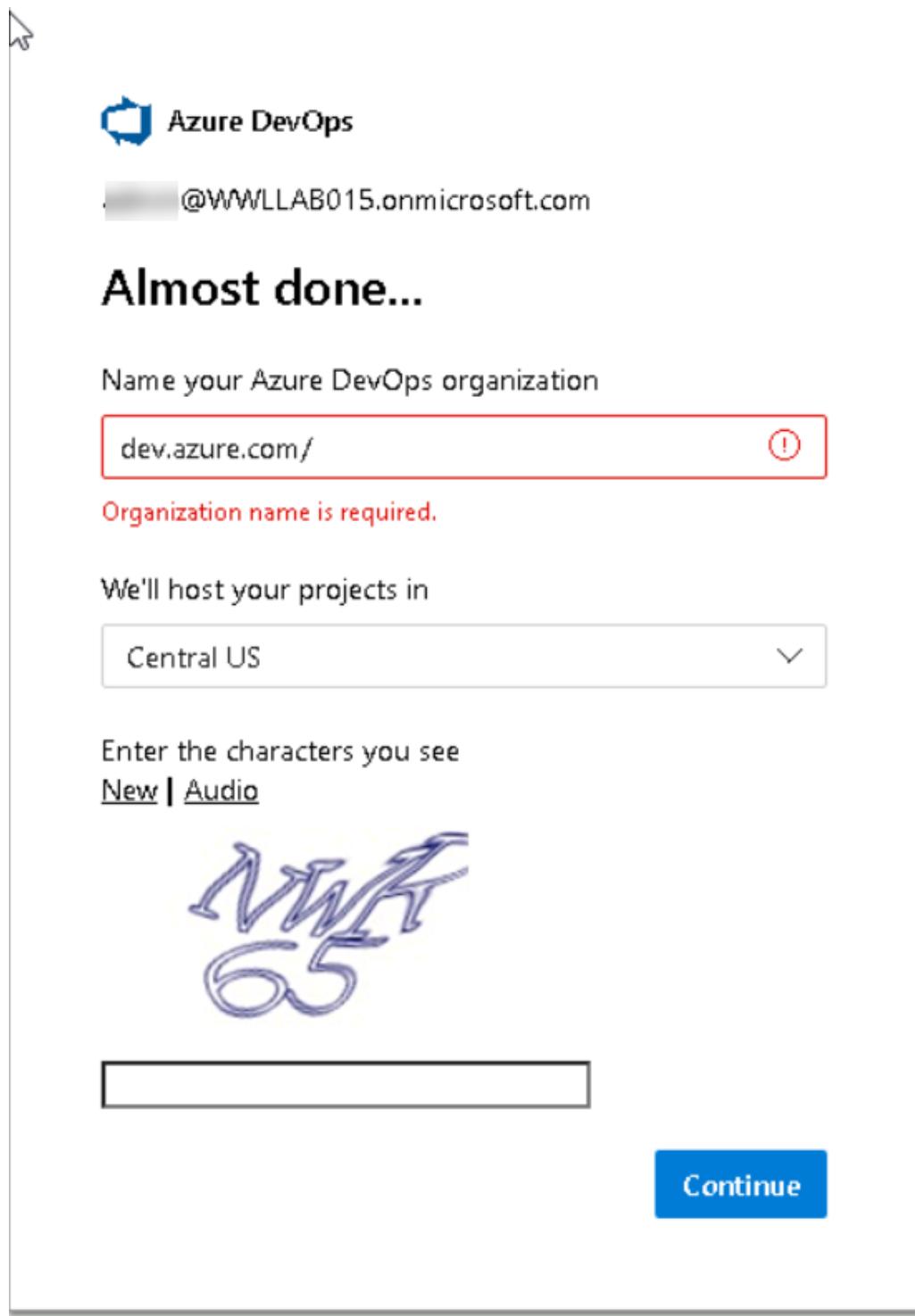
United States

I would like to receive information, tips, and resources related to Microsoft developer tools and services, including Azure DevOps, Visual Studio, Visual Studio Subscriptions, and other Microsoft products and services.

Continue

e. Click **Continue**

f. Click **Continue**



g. Provide a unique Azure DevOps Organization name **Device Management lastnameMMYY** (*replace lastname with your last name, MM current month, and YY with current year.*), select a location closest to your tenant, enter captcha and click **Continue**.

[!NOTE] For some users, the DevOps Organization might be automatically created using your username. This is OK and you do not need to rename the Organization.

[!NOTE] For some users, this page might have the heading "We need a few more details". Ensure that you enter the organization name and not miss this step.

1. Setup billing for Azure DevOps

- a. Choose the newly created organization, and click **Organization settings** in the bottom left-hand corner of the screen



D

New organization

What's new

Sprint 185 release notes

We resolved several dependency tracking issues in Azure Board Delivery Plans 2.0. Check out the release notes to learn more.



Organization settings

b. Click **Billing** and **Setup billing**

- c. Select the Azure Pass subscription and click **Save**
- d. Set **Paid parallel jobs** to **1** for **MS Hosted CI/CD**

Billing

Azure Subscription ID
[REDACTED]

Change billing Configure user billing

Pipelines for private projects	Free	Paid parallel jobs
MS Hosted CI/CD [?]	<input type="text" value="1"/>	<input type="text" value="1"/>
Self-Hosted CI/CD [?]	<input type="text" value="1"/>	<input type="text" value="0"/>

Visit [parallel jobs](#) for full details on free pipelines and public concurrency

- e. Click **SAVE** at the bottom of the screen
 - f. Click on the **Azure DevOps** logo in top left of the screen
2. Create a Azure DevOps project

Projects are containers in Azure DevOps that track work items and source assets. When you set up the automation for the deployment tasks, those will be pipelines built in the context of a project.

Create a project to get started

Project name *

|

Visibility



Public

Anyone on the internet can view the project. Certain features like TFVC are not supported.



Private

Only people you give access to will be able to view this project.

+ Create project

a. Enter **Device Management lastnameMMYY** for **Project Name** and click **Continue**. (*replace lastname with your last name, MM with current month, and YY with current year.*)

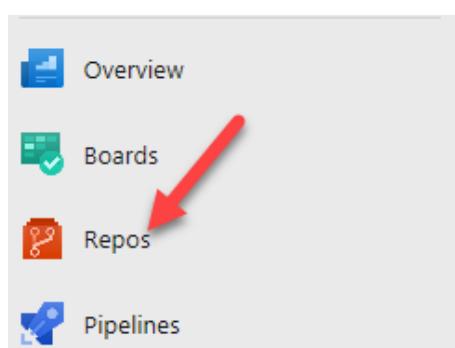
b. Select **Private**

c. Click **Create project**

3. Initialize Repository

An Azure Repo is a source/version control container inside the Azure DevOps project and is used to track changes you make. You will be using it to store the solution files for the team building the Device Ordering app.

a. Select **Repos**



- b. Scroll down, check the **Add a Readme** checkbox, and click **Initialize**
-

Initialize master branch with a README or gitignore

Add a README Add a .gitignore: None ▾ **Initialize**

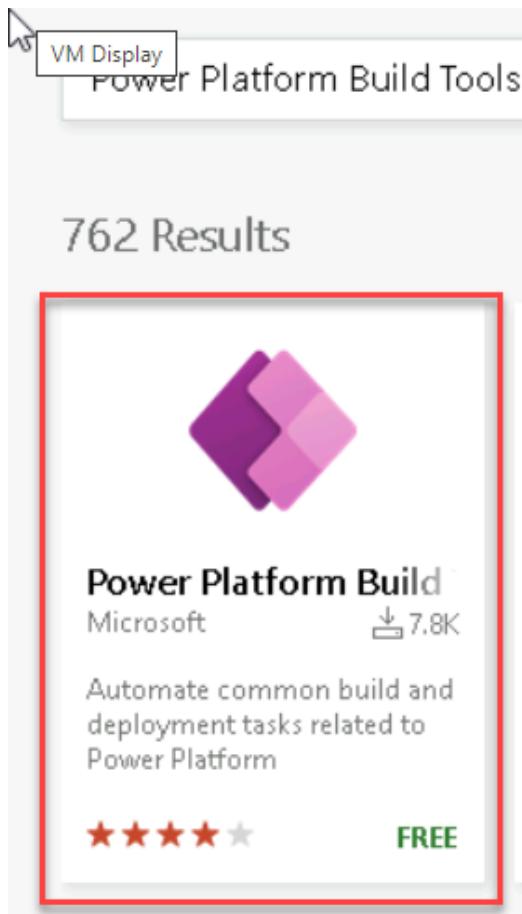
Task 2: Request grant of parallel pipeline jobs

1. If you do not have an Azure Pass, request access to Azure DevOps free tier
 - a. Send an email to azpipelines-freetier@microsoft.com. Include the following information in the email, your name and the name of the Azure DevOps organization: **Device Management lastnameMMYY**

[!NOTE] See <https://devblogs.microsoft.com/devops/change-in-azure-pipelines-grant-for-private-projects/> for more details

Task 3: Configure Power Platform Build Tasks

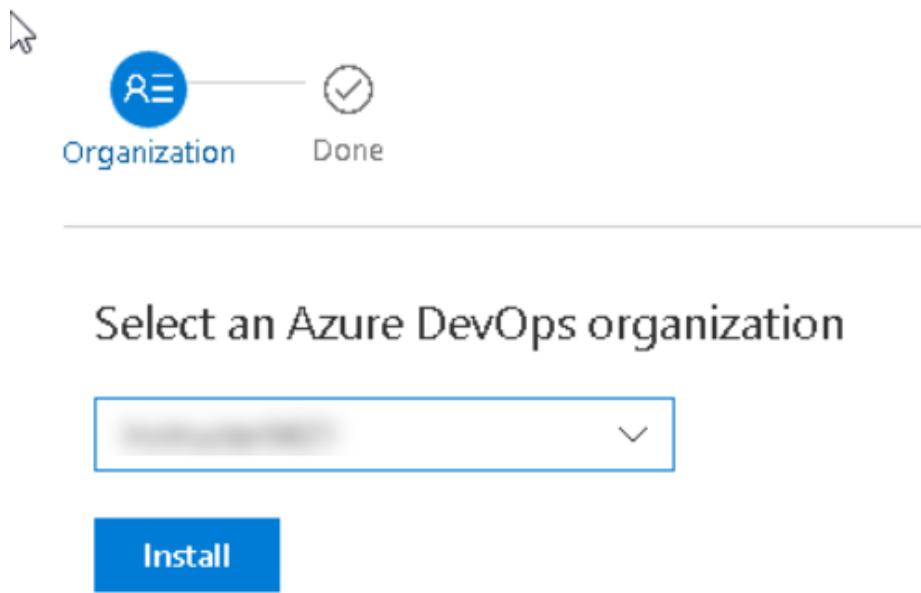
1. Install the Power Platform Build Tools
 - a. Sign in to Visual Studio marketplace
<https://marketplace.visualstudio.com/azuredevops>
 - b. Search for **Power Platform Build Tools**



c. Select Power Platform Build Tools

A screenshot of the Azure DevOps Marketplace showing the 'Power Platform Build Tools (1.0.23)' page. The page header includes the Visual Studio logo and the text 'VisualStudio | Marketplace'. Below the header, the breadcrumb navigation shows 'Azure DevOps > Azure Pipelines > Power Platform Build Tools (1.0.23)'. The main content area features the purple diamond logo, the title 'Power Platform Build Tools (1.0.23)', the developer 'Microsoft', the install count '7,846 installs', the rating '★★★★★ (8)', and the word 'Free'. A description 'Automate common build and deployment tasks related to Power Platform' is also present. A green button labeled 'Get it free' is located at the bottom of the card.

d. Click Get it Free.

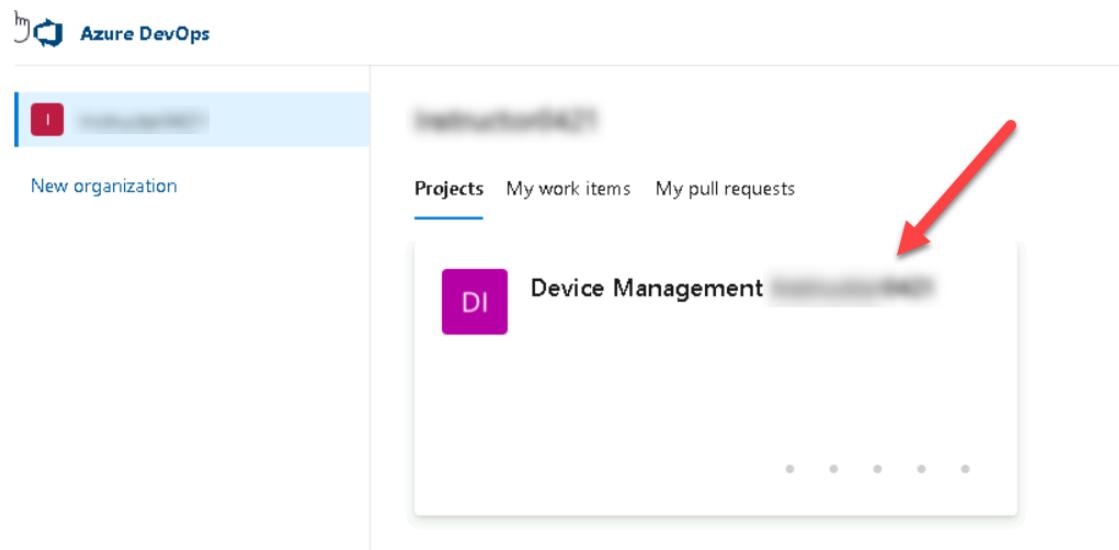


e. Select the **Azure DevOps** organization you created and click **Install**

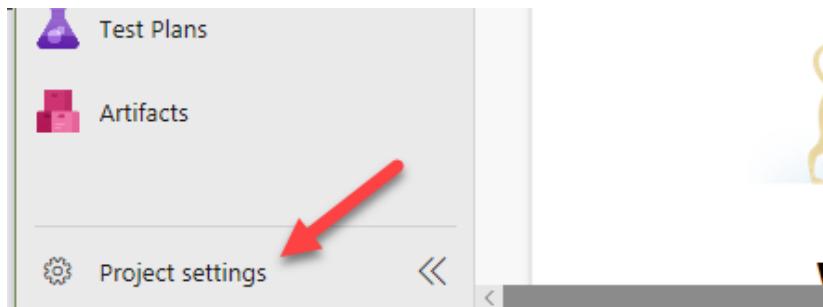
f. Click **Proceed to Organization**

2. Configure Repository security

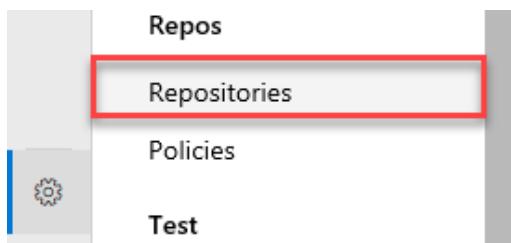
a. Click to open the **Device Management lastnameMMYY** project you created



b. Click **Project Settings** in the bottom left corner of the Azure DevOps



c. Select **Repositories** in the settings pane.



d. Select the **Device Management lastnameMMYY** repo.

A screenshot of the 'All Repositories' page. At the top, there is a navigation bar with tabs: 'Repositories' (underlined in blue), 'Settings', 'Policies', and 'Security'. Below the navigation bar, a list of repositories is shown. One repository, 'Device Management', is highlighted with a red box and has a red arrow pointing from the bottom right towards it. The repository name is preceded by a red diamond icon.

e. Select the **Security** tab

f. Search for **Project Collection Build Service** and select the one not called **Project Collection Build Service Accounts**

Settings Policies Security

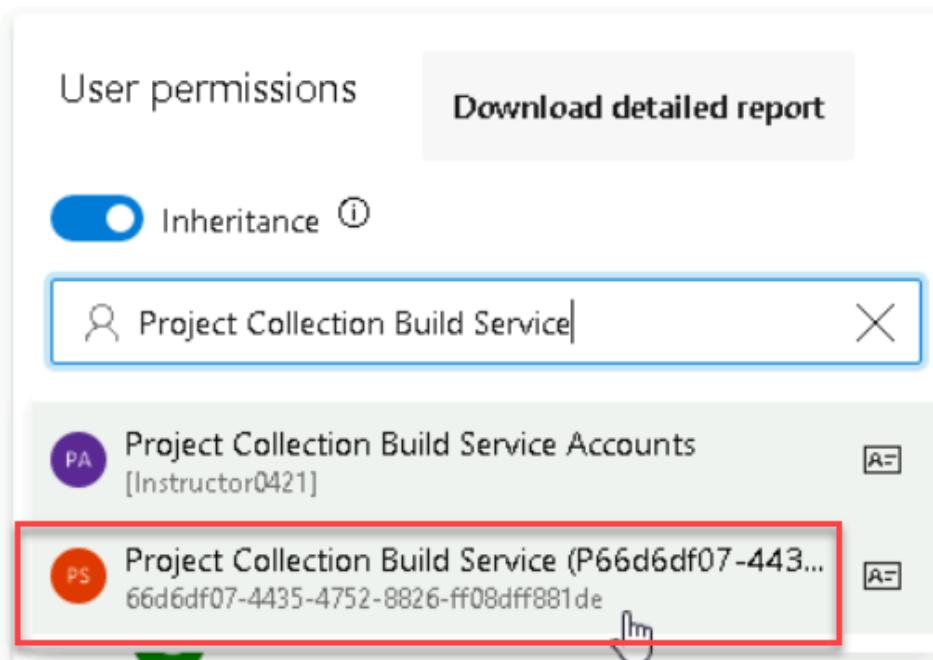
User permissions Download detailed report

Inheritance ⓘ

Project Collection Build Service

Project Collection Build Service Accounts [Instructor0421] ⌂

Project Collection Build Service (P66d6df07-443... 66d6df07-4435-4752-8826-ff08dff881de) ⌂



g. Select **Project Collection Build Service Accounts**

User permissions Download detailed report

Inheritance ⓘ

Project Collection Build Service (P66d6df07-443...)

Bypass policies when completing pull requests Not set

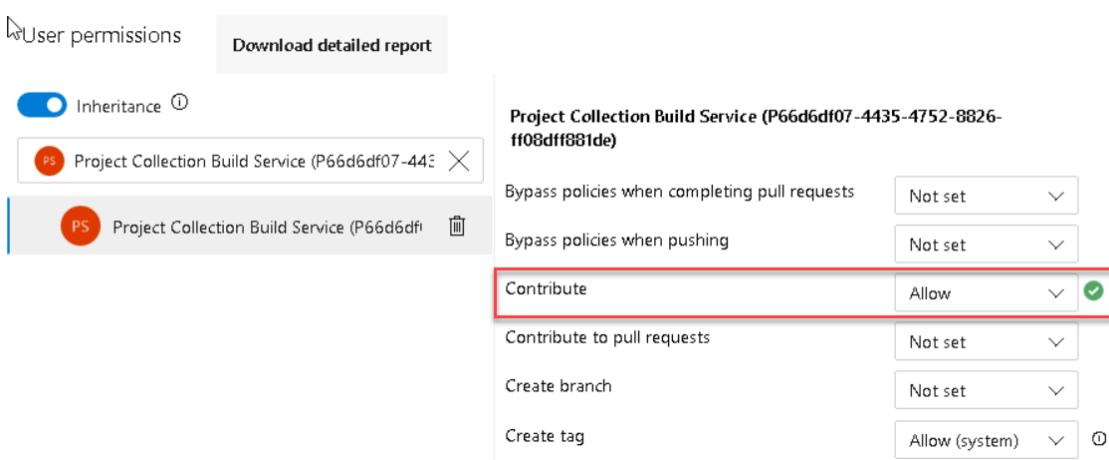
Bypass policies when pushing Not set

Contribute Allow ✓

Contribute to pull requests Not set

Create branch Not set

Create tag Allow (system)



h. Locate **Contribute** and select **Allow**.

i. Click **Show More** to expand the menu.



Repositories

Policies

Test

>>

Retention

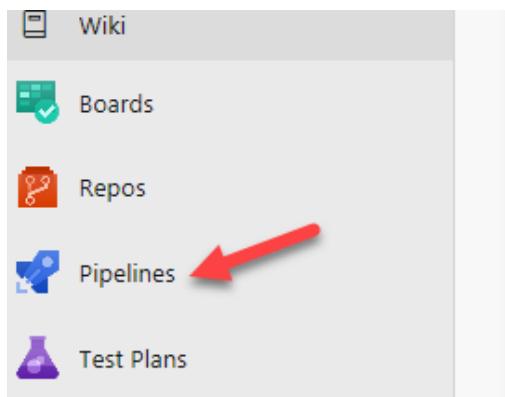
Exercise 4: Export Pipeline

In this exercise, you will build an Azure DevOps pipeline that will export the solution from the development environment, unpack the solution file to individual files and then check those files into the repository. These solution files can then be used to re-create development environments or promote the solution to test/production. You will then run the pipeline.

Task 1: Create a pipeline to export the solution

1. Create a Build Pipeline

- a. Select **Pipelines**



- b. Click **Create Pipeline**.

Create your first Pipeline

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

[Create Pipeline](#)

- c. Click **Use the Classic Editor to create a pipeline without YAML**



Subversion

Centralized version control by Apache



[Use the classic editor to create a pipeline without YAML.](#)

d. Use Azure Repos Git, click **Continue**

Select a source

Azure Repos Git	GitHub	GitHub Enterprise Server	Subversion	Bitbucket Cloud	Other Git
-----------------	--------	--------------------------	------------	-----------------	-----------

Team project

Device Management

Repository

Device Management

Default branch for manual and scheduled builds

main

Continue

e. Select **Empty Job**

Select a template

Or start with an [Empty job](#)

Configuration as code

f. Click **Save & Queue** and select **Save**

Save build pipeline X

Select folder *

\

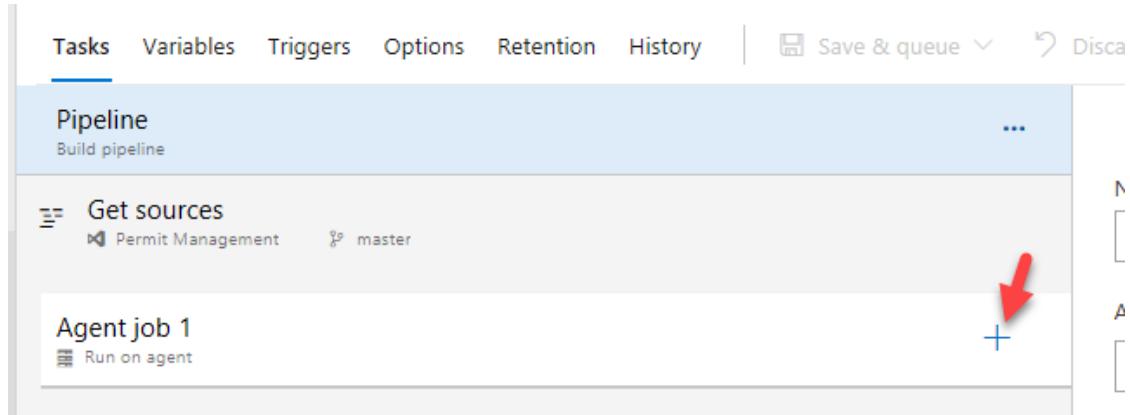
Comment

Save Cancel

g. Click **Save**

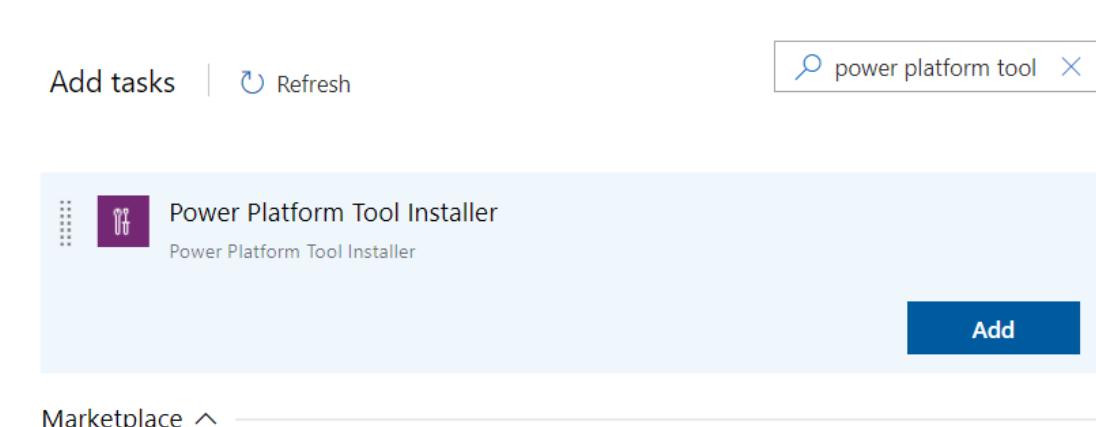
2. Add the Power Platform Tool Installer task

a. Click + icon to add a task **Agent Job 1**



b. Search for **Power Platform Tool**

c. Hover over select **Power Platform Tool Installer**, and click **Add**



3. Add a Power Platform Export Solution task

a. Search for **Power Platform Export**

b. Hover over **Power Platform Export Solution** and click **Add**

4. Open the Power Platform Export Solution task

a. Select the **Power Platform Export Solution** task

The screenshot shows the Azure DevOps Pipeline Editor. A build pipeline named "Agent job 1" is displayed. It contains three tasks: "Power Platform Tool Installer" (Run on agent), "Power Platform Import Solution" (Run on agent), and "Agent job 1" (Run on agent). The "Power Platform Import Solution" task has a red warning icon indicating "Some settings need attention".

5. Create a Service Connection for the development environment

a. With the **Power Platform Export Solution** task selected

The screenshot shows the Azure DevOps Pipeline Editor with the "Power Platform Export Solution" task selected. The task configuration pane on the right includes fields for "Display name" (Power Platform Export Solution), "Authentication type" (Username/password), "Service connection" (selected), "Solution Name" (redacted), "Solution Output File" (redacted), and checkboxes for "Export as Managed Solution" and "Export solution as asynchronous operation". A red arrow points to the "Manage" link next to the "Service connection" dropdown.

b. Click **Manage** next to Service Connection (a new tab should open)

The screenshot shows the "Manage Service Connection" dialog box. It includes fields for "Service connection" (redacted), "Solution Name" (redacted), "Solution Output File" (redacted), and checkboxes for "Export as Managed Solution" and "Export solution as asynchronous operation". A red arrow points to the "Create Service Connection" button at the bottom of the dialog.

c. Click **Create Service Connection**

Choose a service or connection type

Search connection types

- Docker Registry
- Generic
- GitHub
- Python package upload

Learn more Next

- d. Select **Generic** and click **Next**
 - e. Paste the **URL** for the development environment (*the URL should begin with https://*)
 - f. Provide your tenant credentials
 - g. Enter the **Connection Name** as **Dev Connection**
 - h. Click **Save**.
 - i. Close the **Service Connections** browser tab
6. Configure Power Platform Export Solution task to export an unmanaged solution
- a. Go back to the **Build Pipeline** tasks browser tab
 - b. With the **Power Platform Export Solution** task selected
 - c. Locate the **Service Connection** field and click **Refresh**
 - d. Select **Dev Connection**
 - e. Enter **\\$(SolutionName)** for **Solution Name**
 - f. Enter **\\$(Build.ArtifactStagingDirectory)\\$(SolutionName).zip** for **Solution Output File**

Display name *

Authentication type * (i)

Username/password (no MFA support) Service Principal/client secret (supports MFA)

Service connection * (i) | Manage (i)

Dev Connection

Solution Name * (i)

Solution Output File * (i)

 ...

Export as Managed Solution (i)

Advanced (i)

g. Click **Save & Queue** and select **Save**

7. Add a Power Platform Export Solution task to export a managed solution

a. Click + icon to add a task **Agent Job 1**

b. Search for **Power Platform Export**

c. Hover over **Power Platform Export Solution** and click **Add**

d. Select the second **Export Solution** task

e. Select **Dev Connection**

f. Enter **\\${SolutionName}** for **Solution Name**

g. Enter
\\${Build.ArtifactStagingDirectory}\\${SolutionName}_managed.zip
for **Solution Output File**

h. Check **Export as Managed Solution.**

Power Platform Export Solution ⓘ

Link settings View YAML Remove

Task version 0.*

Display name *

Power Platform Export Solution

Authentication type *

Username/password (no MFA support) Service Principal/client secret (supports MFA)

Service connection * ⓘ | Manage ↗

Dev Connection

Solution Name *

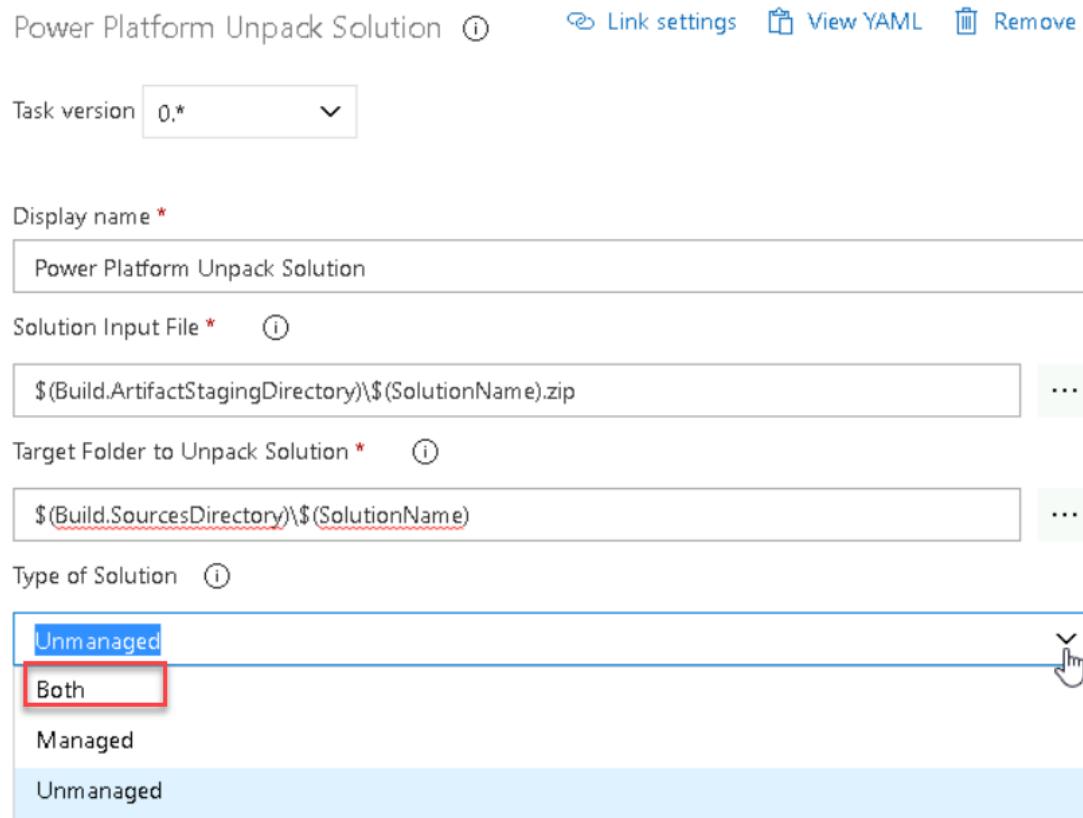
\$(SolutionName)

Solution Output File *

\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip

Export as Managed Solution ⓘ
 Export solution as asynchronous operation ⓘ

- i. Click **Save & Queue** and select **Save** and click **Save**
8. Add an Unpack task to extract files from the solution
 - a. Click + icon to add a task **Agent Job 1**
 - b. Search for **Unpack**
 - c. Hover over **Power Platform Unpack Solution** and click **Add**
 - d. Select the **Unpack** task
 - e. Enter **\\$(Build.ArtifactStagingDirectory)\\$(SolutionName).zip** for **Solution Input File**
 - f. Enter **\\$(Build.SourcesDirectory)\\$(SolutionName)** for **Target Folder**

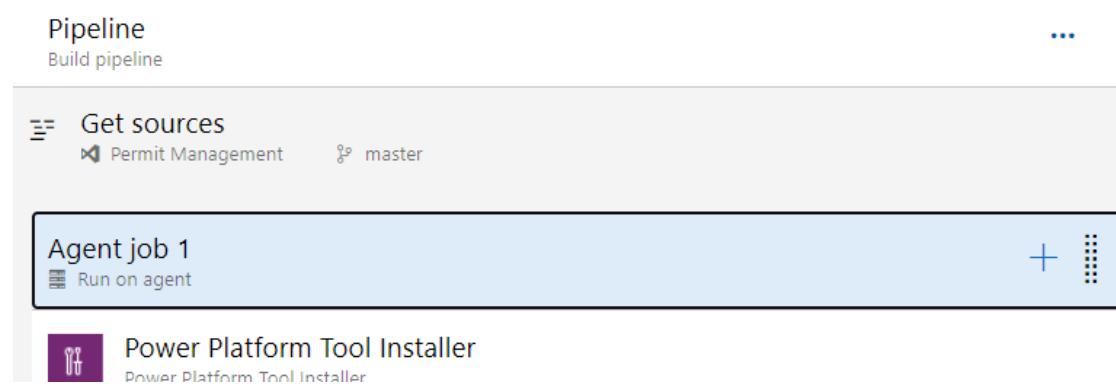


g. Choose **Both** for Type of Solution

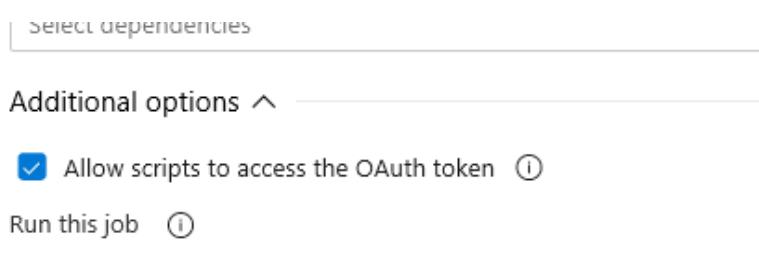
h. Click **Save and Queue** and select **Save** and click **Save**

9. Allow scripts to access the OAuth Token

a. Select **Agent Job 1**



b. Scroll down and check the **Allow Scripts to Access the OAuth Token** checkbox



c. Click **Save & Queue** and select **Save**

d. Click **Save**

[!IMPORTANT] This will allow the commands in the pipeline to check in files to the Azure DevOps repo to work. **This is a very important step which if skipped will cause the pipeline to fail**

1. Add Command Line task to check in the solution changes to the repo

a. Click + icon to add a task **Agent Job 1**

b. Search for **Command Line**

c. Hover over **Command Line** and click **Add**

d. Select the Command Line task

e. Paste the script below in the **Script** text area

```
echo commit all changes git config user.email  
"user@myorg.onmicrosoft.com" git config user.name "Automatic  
Build" git checkout main git add --all git commit -m  
"solution updates" echo push code to new repo git -c  
http.extraheader="AUTHORIZATION: bearer  
\$(System.AccessToken)" push origin main
```

Replace **user\@myorg.onmicrosoft.com** with your tenant credentials

Display name *

Script * (i)

```
echo commit all changes
git config user.email "████████@WWLLAB.████.onmicrosoft.com"
git config user.name "Automatic Build"
git checkout main
git add --all
git commit -m "solution updates"
echo push code to new repo
git -c http.extraheader="AUTHORIZATION: bearer $(System.AccessToken)" push origin main
```

2. Add Solution Name variable

a. Select the **Variables** tab

b. Click **+ Add**.

The screenshot shows the 'Variables' tab selected in the top navigation bar. Below it, the 'Pipeline variables' section lists several system variables with their values. A red arrow points to the '+ Add' button at the bottom left of the table.

Name ↑	Value
system.collectionId	794a909a-8e15-45b6-8f90-cc1dd641
system.debug	false
system.definitionId	1
system.teamProject	Permit Management

c. Enter **SolutionName** for Name

d. Enter **ContosoDeviceOrderManagement** for Value

The screenshot shows the 'Variables' tab selected. The 'Pipeline variables' table now includes a new row for 'SolutionName' with the value 'ContosoDeviceOrderManagement'. This row is highlighted with a red box.

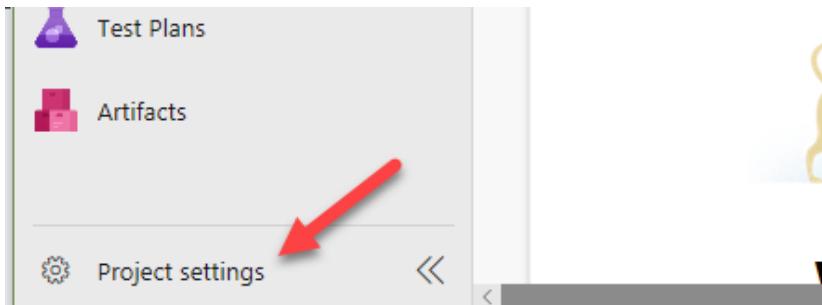
Name ↑	Value
system.collectionId	66d6df07-4435-4752-8826-ff08dff881de
system.debug	false
system.definitionId	1
system.teamProject	Device Management Instructor0421
SolutionName	ContosoDeviceOrderManagement

e. Click **Save & Queue** and select **Save**

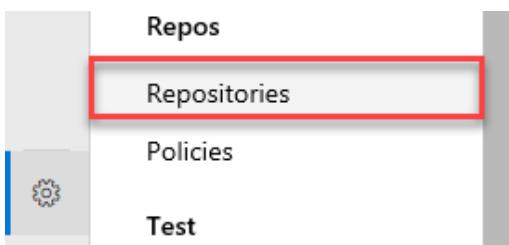
f. Click **Save** again

3. Configure Repository security

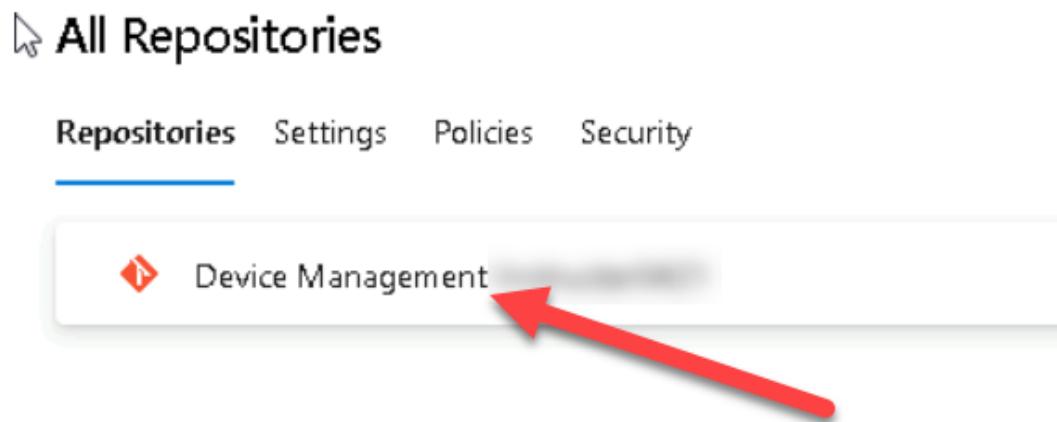
- Click **Project Settings** in the bottom left corner of the Azure DevOps



- Select **Repositories** in the settings pane.

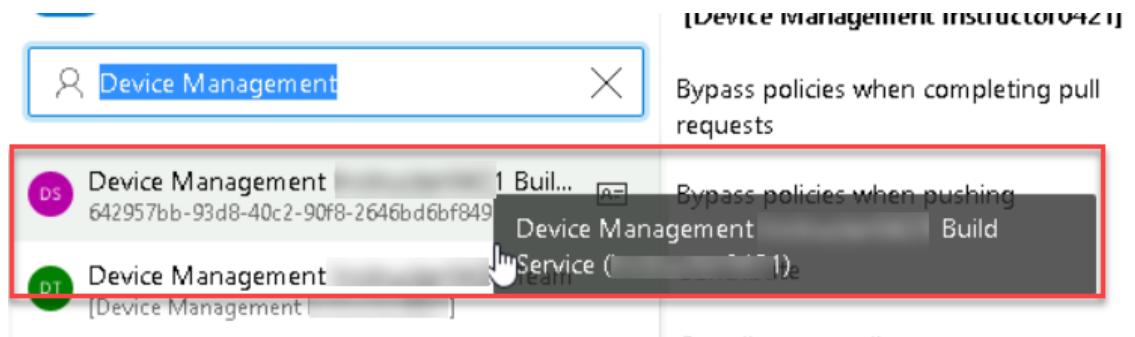


- Select the **Device Management lastnameMMYY** repo.



- Select the **Security** tab

- Search for **Device Management lastnameMMYY Build Service** and select **Device Management lastnameMMYY Build Service user**



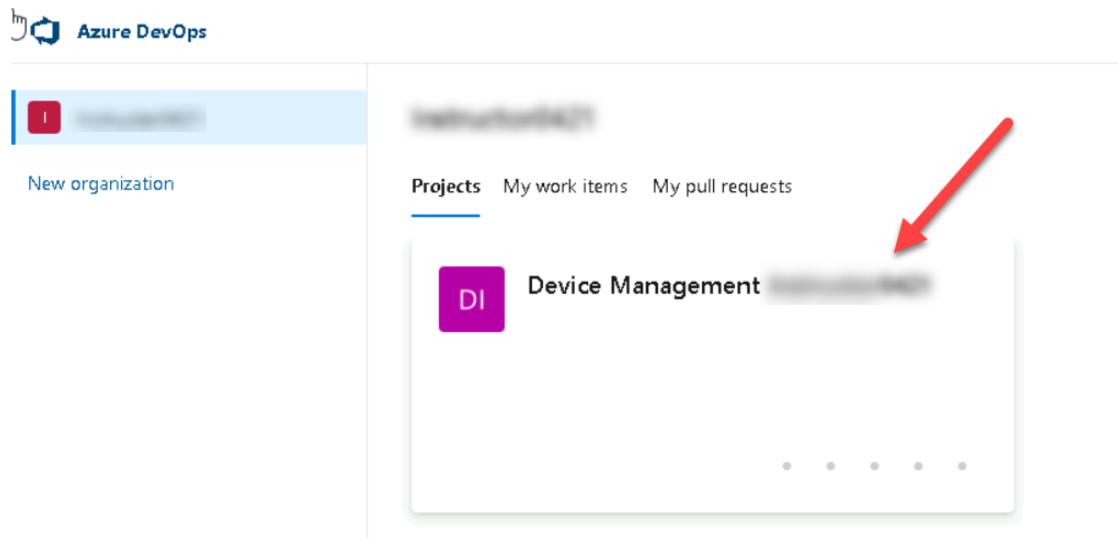
f. Select Project Collection Build Service Accounts

The screenshot shows the 'Management Instructor0421' project settings in Azure DevOps. The 'Security' tab is selected. A red box highlights the 'Contribute' permission row for the 'Device Management' service account. The 'Allow' checkbox is checked, and a green checkmark is visible.

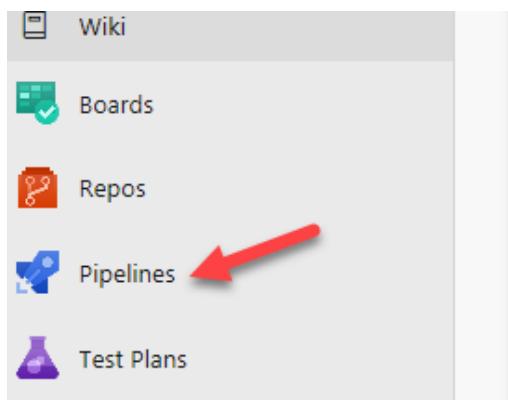
g. Locate Contribute and select Allow.

Task 2: Run the Pipeline

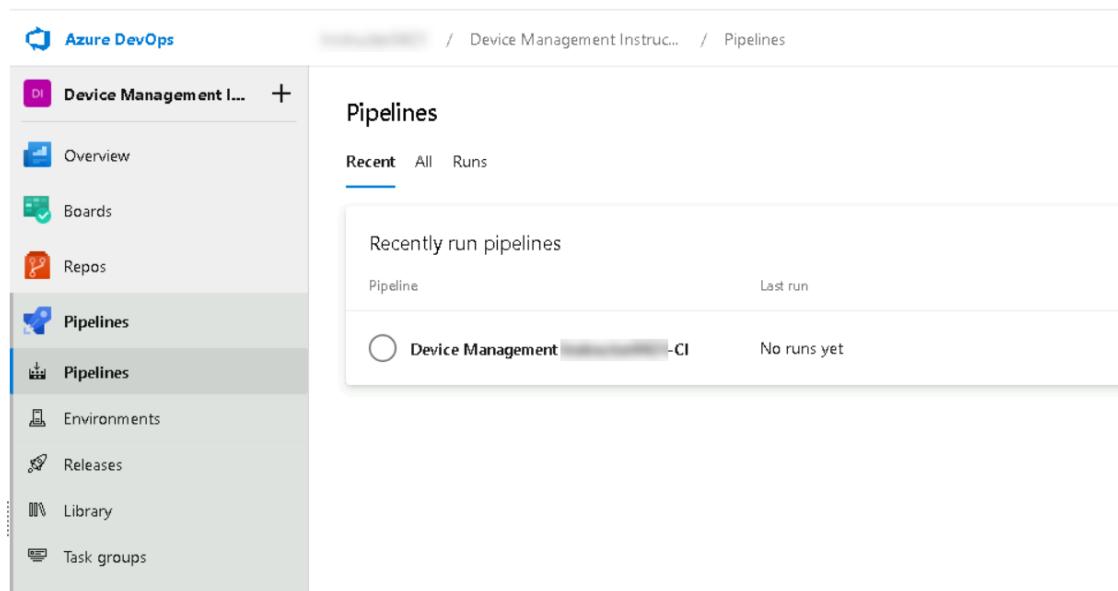
1. Open the build pipeline
 - a. Navigate to Azure Devops <https://dev.azure.com>
 - b. Click to open the **Device Management lastnameMMYY** project



c. Select Pipelines



d. Click to open the Device Management lastnameMMYY-CI pipeline



e. Click Run pipeline

f. Click Run

The screenshot shows the 'Summary' page for a pull request titled '#1 Added README.md'. At the top, there's a 'Cancel' button and a three-dot menu icon. Below that, the 'Summary' section includes details like 'Manually run by [user]' and 'Repository and version' (Device Management). It also shows 'Time started and elapsed' (Just now), 'Related' items (0 work items), and 'Tests and coverage' (Get started). The 'Jobs' section lists 'Agent job 1' with a status of 'Queued'. A 'View change' button is located in the top right corner.

g. Wait for the job to complete

The screenshot shows the 'Jobs' table. The first row has a header with columns 'Name', 'Status', and 'Duration'. The second row contains the data for 'Agent job 1': it is marked as 'Success' and took '3m 38s'. A red arrow points to the 'Agent job 1' row.

Name	Status	Duration
Agent job 1	Success	3m 38s

h. Click to open the job

Jobs		
▼	✓ Agent job 1	3m 38s
	Initialize job	2s
	Checkout Permit Mana...	7s
	Power Platform Tool In...	29s
	Power Platform Ex...	2m 48s
	Power Platform Unpack...	5s
	Command Line Script	4s
	Post-job: Checkout Pe...	<1s
	Finalize Job	<1s
	Report build status	<1s

2. Review the Repository

- a. Select Repos
- b. Click to open the **ContosoDeviceOrderManagement** folder
- c. Examine the content of each folder.

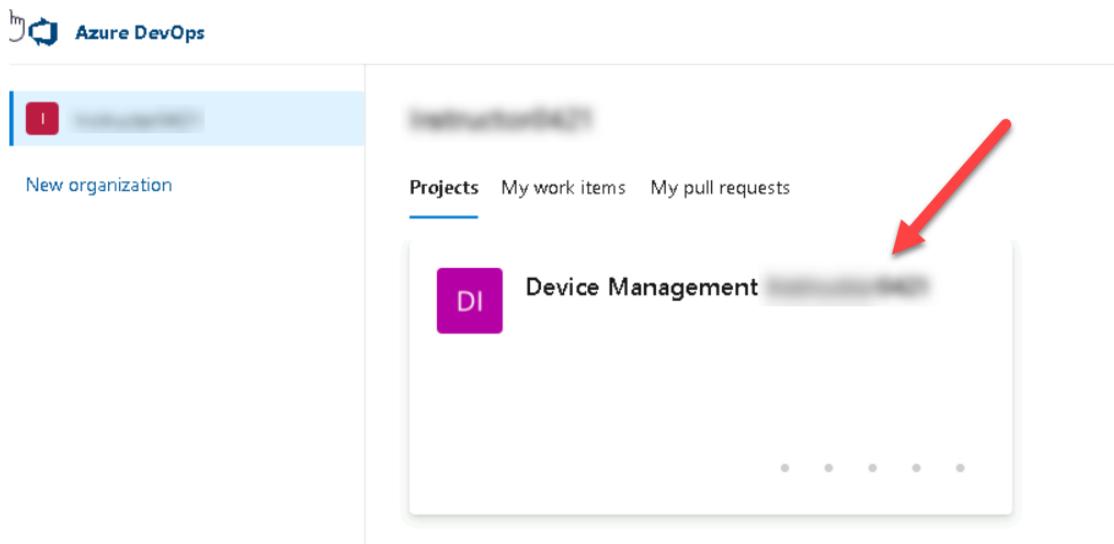
Exercise 5: Build managed solution and publish artifacts

In this exercise, you will take the solution files checked into the repo in the previous steps and re-package them into a managed solution file. This solution file will be published as a build artifact so it can be used in the release pipeline that we are going to use to publish to test and production.

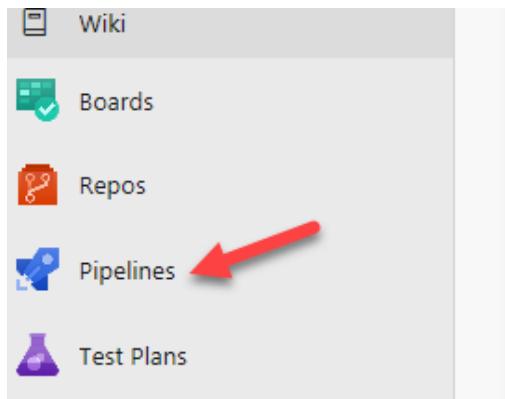
In a real project this is where you could add steps to import the solution into a build Dataverse environment to check for missing dependencies. You could also add build tasks to run tests against your solution as well as run Power Platform Solution Checker to detect problems. In this lab exercise we will skip those extra steps to ensure you have enough time to complete the lab.

Task 1: Build the Managed Solution

1. Create a pipeline to build the managed solution from the repo
 - a. Navigate to Azure Devops <https://dev.azure.com>
 - b. Click to open the **Device Management lastnameMMYY** project

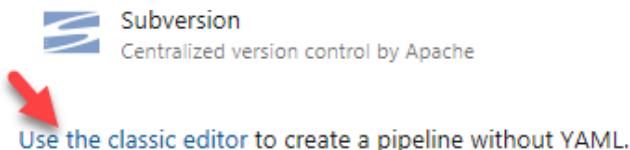


- c. Select **Pipelines**



d. Click **New Pipeline**

c. Click **Use the Classic Editor to create a pipeline without YAML**



d. Use Azure Repos Git, click **Continue**

Select a source

Azure Repos Git	GitHub	GitHub Enterprise Server	Subversion	Bitbucket Cloud	Other Git
-----------------	--------	--------------------------	------------	-----------------	-----------

Team project

Device Management

Repository

Device Management

Default branch for manual and scheduled builds

main

Continue

The form consists of several dropdown menus and buttons. At the top, there's a 'Select a source' section with a grid of icons for different version control systems. The 'Azure Repos Git' option is selected and highlighted with a blue border. Below this, there are sections for 'Team project' (containing a dropdown with 'Device Management'), 'Repository' (containing a dropdown with 'Device Management'), and 'Default branch for manual and scheduled builds' (containing a dropdown with 'main'). At the bottom right, there's a large green button labeled 'Continue'.

e. Select **Empty Job**



- f. Enter **Build Managed Solution** for **Name**
- f. Click **Save & Queue** and select **Save**
- g. Click **Save**
2. Add the Power Platform Tool Installer task
 - a. Click + icon to add a task **Agent Job 1**
 - b. Search for **Power Platform Tool**
 - c. Hover over select **Power Platform Tool Installer**, and click **Add**
3. Add a Pack task to create the solution
 - a. Click + icon to add a task **Agent Job 1**
 - b. Search for **Pack**
 - c. Hover over **Power Platform Pack Solution** and click **Add**
 - d. Select the **Pack** task
 - e. Enter `\$(Build.SourcesDirectory)\$(SolutionName)` for **Source Folder of Solution to Pack**
 - f. Enter `\$(Build.ArtifactStagingDirectory)\$(SolutionName)_managed.zip` for **Solution Output Folder**
 - g. select **Managed** for **Type of Solution**

Power Platform Pack Solution ⓘ

Link settings View YAML Remove

Task version 0.*

Display name *

Power Platform Pack Solution

Source Folder of Solution to Pack * ⓘ

`$(Build.SourcesDirectory)\$(SolutionName)`

Solution Output File * ⓘ

`$(Build.ArtifactStagingDirectory)\$(SolutionName)_managed.zip`

Type of Solution ⓘ

Managed

h. Click **Save & Queue** and select **Save** and click **Save**

4. Add Solution Name variable

a. Select the **Variables** tab

b. Click **+ Add**.

Tasks	Variables	Triggers	Options	Retention	History	Save & queue	Discard	Summary	Queue	...
Pipeline variables										
Variable groups										
Predefined variables ↗										
						Name ↑				Value
						system.collectionId				794a909a-8e15-45b6-8f90-cc1dd641
						system.debug				false
						system.definitionId				1
						system.teamProject				Permit Management
						+ Add				

c. Enter **SolutionName** for **Name**

d. Enter **ContosoDeviceOrderManagement** for **Value**

The screenshot shows the 'Variables' tab in the Azure Pipelines interface. It lists several system variables and a user-defined variable 'SolutionName'. The 'SolutionName' row is highlighted with a red box.

Name	Value
system.collectionId	66d6df07-4435-4752-8826-ff08dff881de
system.debug	false
system.definitionId	1
system.teamProject	Device Management Instructor0421
SolutionName	ContosoDeviceOrderManagement

e. Click **Save & Queue** and select **Save**

f. Click **Save** again

5. Add a Publish task for the solution

a. Click + icon to add a task **Agent Job 1**

b. Search for **Publish Pipeline**

c. Hover over **Publish Pipeline Artifacts** and click **Add**

d. Select the **Publish Pipeline Artifacts** task

e. Enter

\\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip
for File or directory Path

f. Enter **drop** for **Artifact Name**

The screenshot shows the 'Tasks' tab in the Azure Pipelines interface. A 'Publish Pipeline Artifacts' task is selected. The configuration includes:

- Task version: 1.*
- Display name: Publish Pipeline Artifact
- File or directory path: \$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip
- Artifact name: drop
- Artifact publish location: Azure Pipelines

g. Click **Save & Queue** and select **Save & Queue** and click **Save and Run**

h. Wait for the run to complete

i. Click to open the job

The screenshot shows two panels. The left panel is titled 'Jobs in run #12' for a 'Build Managed Solution'. It lists several tasks: 'Agent job 1' (duration 43s), 'Initialize job' (2s), 'Checkout Device Mana...' (8s), 'Power Platform Tool In...' (25s), 'Power Platform Pack Sol...' (4s), 'Publish Pipeline Artifact' (3s), 'Post-job: Checkout De...' (<1s), 'Finalize Job' (<1s), and 'Report build status' (<1s). The 'Agent job 1' row is highlighted with a red box. The right panel is titled 'Agent job 1' and displays its configuration: Pool: Azure Pipelines, Image: vs2017-win2016, Agent: Hosted Agent, Started: Just now, Duration: 43s. It also shows the steps: Job preparation parameters, which resulted in 1 queue time variable used and 1 artifact produced.

j. Click on the **Artifact** link

The screenshot shows the 'Artifacts' page with the 'Published' tab selected. It lists two artifacts: 'drop' (size 112 KB) and 'ContosoDeviceOrderManagement_managed.zip' (size 112 KB). The 'drop' folder is expanded, showing its contents.

k. Expand the **drop** folder and you should see the managed solution

[!NOTE] The artifact link can be found by selecting **Agent job 1**. If you do not see the artifacts, refresh the browser and it will appear.

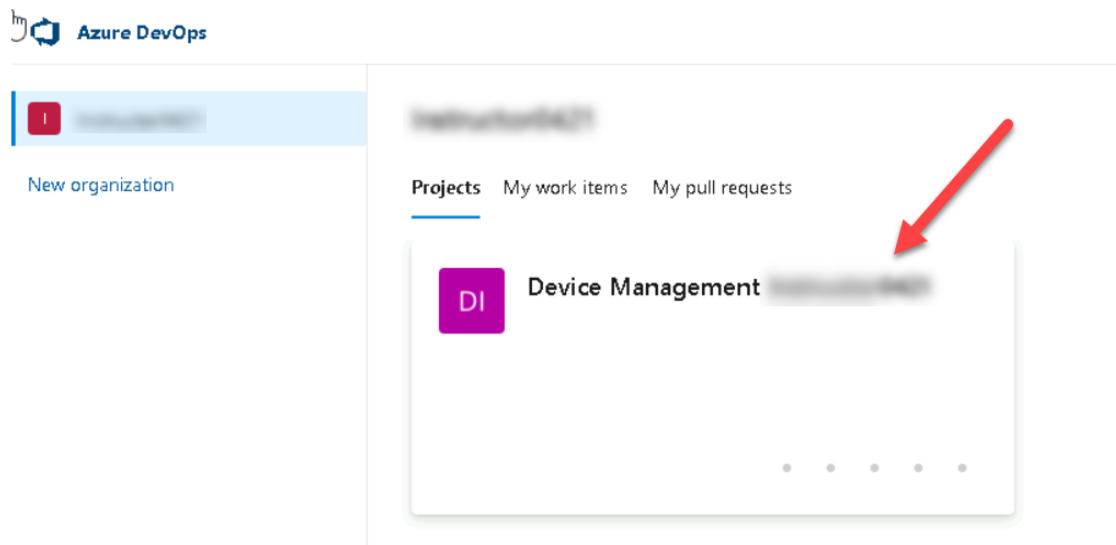
Exercise 6: Release Pipeline

In this exercise, you will build a release pipeline. The release pipeline is intended to take the output from the build pipeline and coordinate deployments to one or more release environments. A common release pipeline might deploy to dev -> test -> user acceptance -> production. Release pipelines can have approval requirements between each environment.

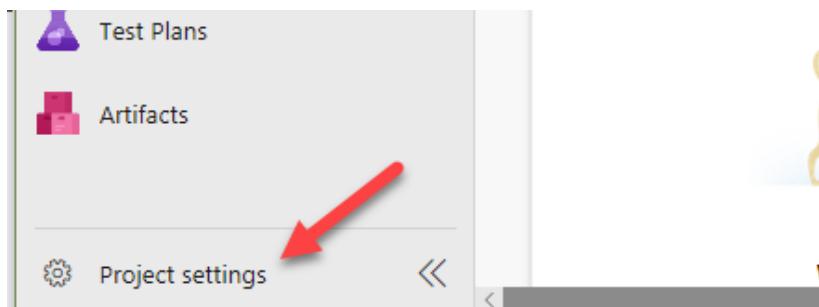
For the purposes of this lab we are only going to deploy to one production environment.

Task 1: Create Connection for Prod

1. Create a Service Connection for the production environment
 - a. Navigate to Azure Devops <https://dev.azure.com>
 - b. Click to open the **Device Management lastnameMMYY** project



- b. Click **Project Settings** in the bottom left corner of the Azure DevOps



- c. Click **Service Connections**
- d. Click **New service connection**.
- e. Select **Generic** and click **Next**.

Choose a service or connection type

Search connection types

- Docker Registry
-  Generic
-  GitHub
-  Python package upload

Learn more Next

- f. Paste the **URL** for the production environment (*the URL should begin with https://*)
- g. Provide your tenant credentials
- h. Enter the **Connection Name** as **Prod Connection**
- i. Click **Save**

Task 2: Create a Release pipeline to import the solution to production

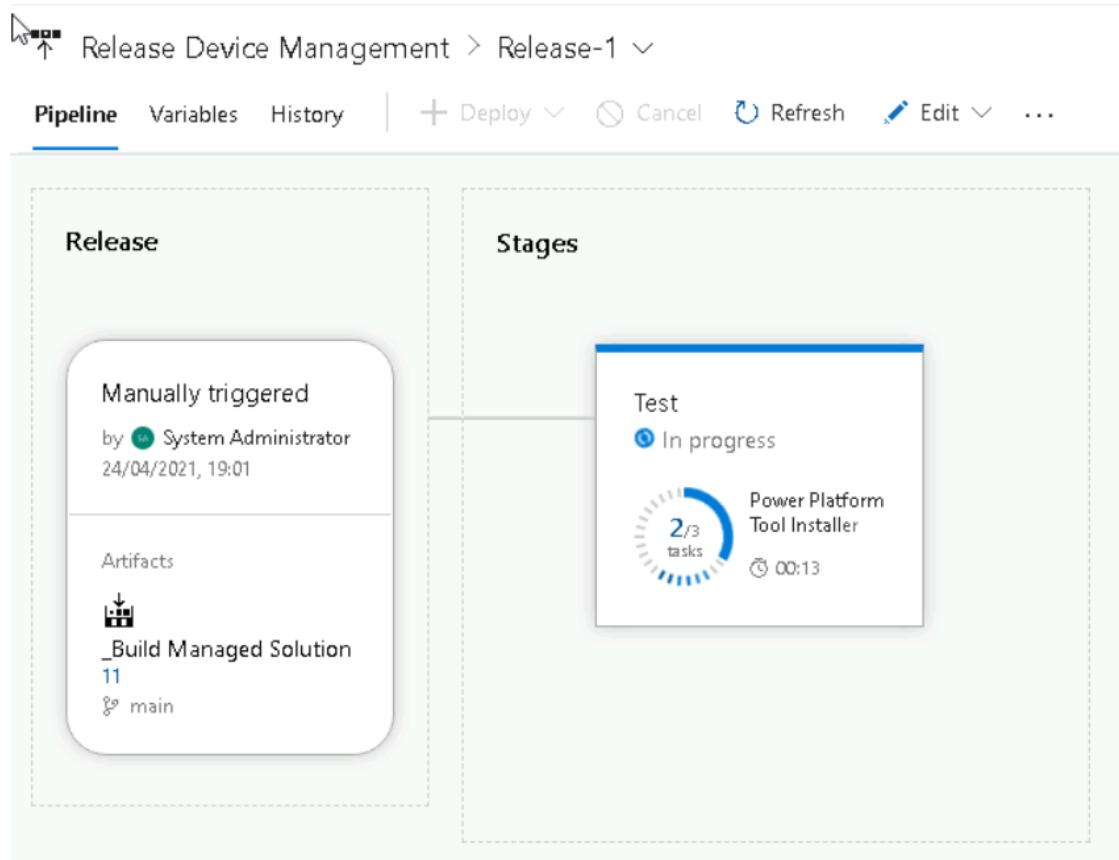
- 1. Create a Release pipeline
 - a. In the **Device Management lastnameMMYY** project
 - b. Select **Pipelines -> Releases**
 - c. Click **New pipeline**
 - d. Select **Empty Job**

- e. Enter **Test** for **Stage Name**
 - f. Click **Save**
 - g. Click **Save** again
2. Add an artifact
- a. click + **Add** under **Artifacts**
 - b. Select **Build Managed Solution** for **Source** and click **Add**
3. Add the Power Platform Tool Installer task
- a. Select the **Tasks** tab
 - b. Click + icon to add a task **Agent job**
 - c. Search for **Power Platform Tool**
 - d. Hover over select **Power Platform Tool Installer**, and click **Add**
4. Add an Power Platform Import Solution task
- a. Click + icon to add a task **Agent job**
 - b. Search for **Power Platform Import**
 - c. Hover over **Power Platform Import Solution**, and click **Add**
5. Configure the **Power Platform Import Solution** task
- a. Select the **Import Solution** task
 - e. Select **Prod Connection**
 - f. Enter **\\$(System.DefaultWorkingDirectory)/_Build Managed Solution/drop\\$(SolutionName)_managed.zip** for **Solution Input File**
 - g. Enter **\\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip** for **Solution Output File**
 - i. Click **Save** and click **OK**
- [!NOTE] There are spaces around the word Managed.

1. Add Solution Name variable
 - a. Select the **Variables** tab
 - b. Click **+ Add**.
 - c. Enter **SolutionName** for **Name**
 - d. Enter **ContosoDeviceOrderManagement** for **Value**
 - e. Click **Save** and click **OK**

Task 3: Release the solution to production

1. Run the release pipeline
 - a. Click **Create Release**
 - b. Click **Create**
 - c. Select **Release-1** and click to open the release



- d. Wait for the release tasks to complete

2. Verify the managed solution has been imported

a. Log on to <https://make.powerapps.com>

b. Select your **Prod** environment

c. Select **Solutions**

You should see the **Contoso Device Order Management** solution as a managed solution.

d. Click to open the **Contoso Device Order Management** solution

You should see the **You cannot directly edit the components within a managed solution** notification message

e. Select **Apps**

You should see Device Ordering App Canvas app and the Device Procurement Model-driven app.

lab: title: 'Lab 02: App composition and solution segmentation'

Module 8: App composition and solution segmentation

Lab Scenario

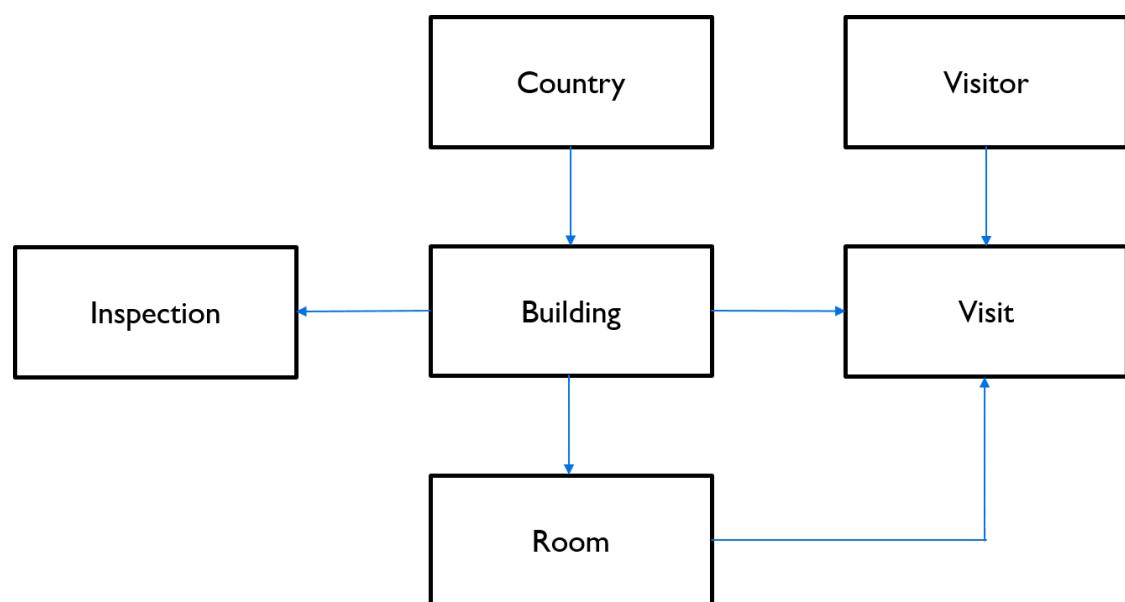
In this hands-on lab, you are a solution architect for Contoso.

A solution is required to manage building security:

- Tracking visitors
- Access control
- Parking passes
- Inspections
- Security staff shifts
- Incidents

A Minimal Viable Product (PVP) has been developed for Phase 1, visitor tracking, in the Power Platform.

The entity relationship diagram for the MVP is as follows:



The contact table has been used for the Visitor entity.

All components are in a single solution.

High Level Steps

In this lab you will need to:

- Define the apps that need to be built
- Determine the number of solutions and how solutions will be layered

Lab Test Environment

You will be assigned one or more users to use to complete the tasks. As this is a shared environment, some tasks that require a tenant Global Administrator or a Service Administrator role will have already been performed.

Exercise 1: App composition

In this exercise you will define which apps are required and which app type should be used.

Requirements

- A security officer must be able to see the visitors due to arrive in each building for the current day
- For each visitor, the system must track the date, building, and employee responsible for the visitor
- Each visitor must be able to check in and check out using a tablet app at the public entrances of each building
- A security officer must be able to see who the visitor is meeting and their expected entry and exit times
- A security officer must be able to quickly and easily record when a visit arrives and leaves the building
- A security officer must be able to see if a visitor has not left by the scheduled exit time
- All visitor records should be deleted automatically after 90 days
- A security manager needs to see the volume of planned visitors by building so they can plan the security staff appropriately
- A security manager needs to be able to find which employees are booking visits
- Employees must be able to book visits easily for buildings they are entitled to use
- A code should be generated for each visit, that code should be available as a QR code for easy entry and exit
- Visitors must be able to access their visit details and enter their car registration to book a parking place
- Visitor parking is very limited at each building. Parking is booked on first come first served basis.
- The system should not allow visitors to book if there are no available places.
- Access to buildings for employees is controlled by a swipe card entry system. This system is hard to extract information from and can only be accessed by one dedicated computer in each building.
- The new Security System should track the buildings each employee has access to and manage adding new employees and new entitlements
- The Security System must automatically update the information in the swipe card entry system.
- A security manager needs to schedule regular inspections of the buildings and assign inspections to security officers.

- A security office must be able to see their list of inspections and to record the results of those inspections.
- The system needs to record security related incidents. All employees should be able to raise an incident. The security staff must be able to review and respond to these incidents.
- Some security staff are not employees of the company; they are sub-contracted workers.

Task 1: Apps

1. Review the apps in the MVP solution
2. Determine what apps will be required for the full business solution
3. Decide which app type will be used for each app and who will use each app

Exercise 2: Solution segmentation

In this exercise you will determine how many solutions are required and how the solutions are to be segmented.

Task 1: Plan solution segmentation

1. Review the components in the MVP solution
2. Estimate the number and type of other components that will be required for the c
3. Determine if you need to segment the solution and if you should use horizontal splitting or vertical layering
4. Map the components onto each solution

Task 2: Solutions

1. Create the solutions in your environment
 2. Add the components from the MVP solution to your solutions
-

lab: title: 'Lab 03: Robotic Process Automation'

Module 15: Robotic Process Automation

Lab Scenario

In this hands-on lab, you are a solution architect for Contoso, helping them automate their legacy applications.

You need to integrate the legacy Windows Contoso Invoicing application in an end-to-end workflow.

In this lab, you will be using Power Automate Desktop to create and run a desktop flow.

High Level Steps

In this lab you will configure Power Automate Desktop to run on your computer, create a desktop flow for a legacy Windows application, and create a cloud flow to retrieve data from Dataverse and enter that data into the legacy application.

Lab Test Environment

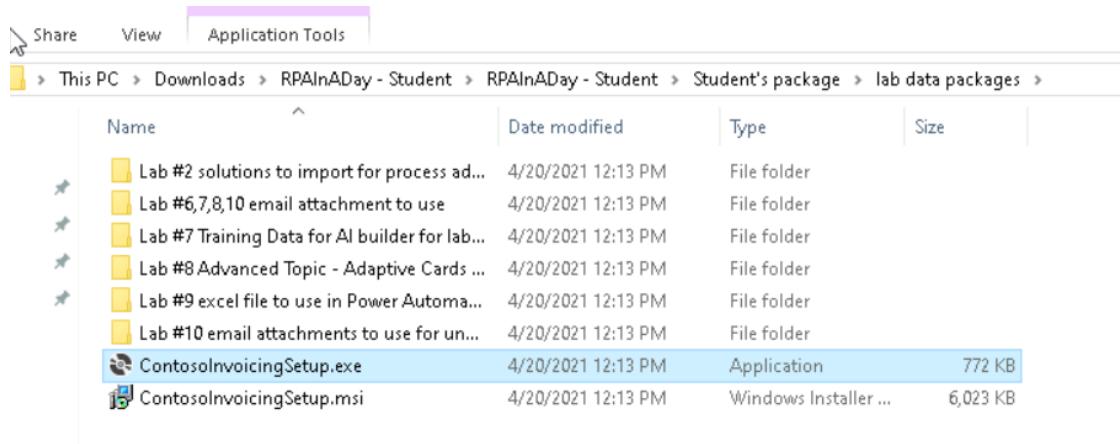
You will be assigned one or more users to use to complete the tasks. As this is a shared environment, some tasks that require a tenant Global Administrator or a Service Administrator role will have already been performed.

Exercise 1: Initialize Software

In this exercise you will install the legacy invoicing application and Power Automate software.

Task 1: Contoso Invoicing

1. Open a browser and navigate to <https://aka.ms/RPAinaDayPackage> **Note:**
A Zip file will be downloaded
2. Extract the files from the Zip file
3. Open the **lab data file packages** folder



Name	Date modified	Type	Size
Lab #2 solutions to import for process ad...	4/20/2021 12:13 PM	File folder	
Lab #6,7,8,10 email attachment to use	4/20/2021 12:13 PM	File folder	
Lab #7 Training Data for AI builder for lab...	4/20/2021 12:13 PM	File folder	
Lab #8 Advanced Topic - Adaptive Cards ...	4/20/2021 12:13 PM	File folder	
Lab #9 excel file to use in Power Automa...	4/20/2021 12:13 PM	File folder	
Lab #10 email attachments to use for un...	4/20/2021 12:13 PM	File folder	
ContosoInvoicingSetup.exe	4/20/2021 12:13 PM	Application	772 KB
ContosoInvoicingSetup.msi	4/20/2021 12:13 PM	Windows Installer ...	6,023 KB

1. Run **ContosoInvoicingSetup.exe**
2. Complete the installation with the defaults

Task 2: Power Automate Desktop

1. Navigate to <https://flow.microsoft.com>
2. Click **Sign in**
3. Provide your tenant credentials and click **Next**.
4. Provide your password and click **Sign in**
5. Click **Yes** **Note:** If prompted, click Get Started
6. Select the production environment

The screenshot shows the Microsoft Power Automate interface. At the top, there's a blue header bar with icons for environments (WWLLAB015), settings, help, and user profile (SA). Below the header, the title "Environments" is displayed. A search bar with a magnifying glass icon and the placeholder "Search" is present. Under the title, the text "Contoso (Default)" is shown. Below it, a list item "✓ WWLLAB015" is highlighted with a red rectangular border. To the right of the list item is a close button (X).

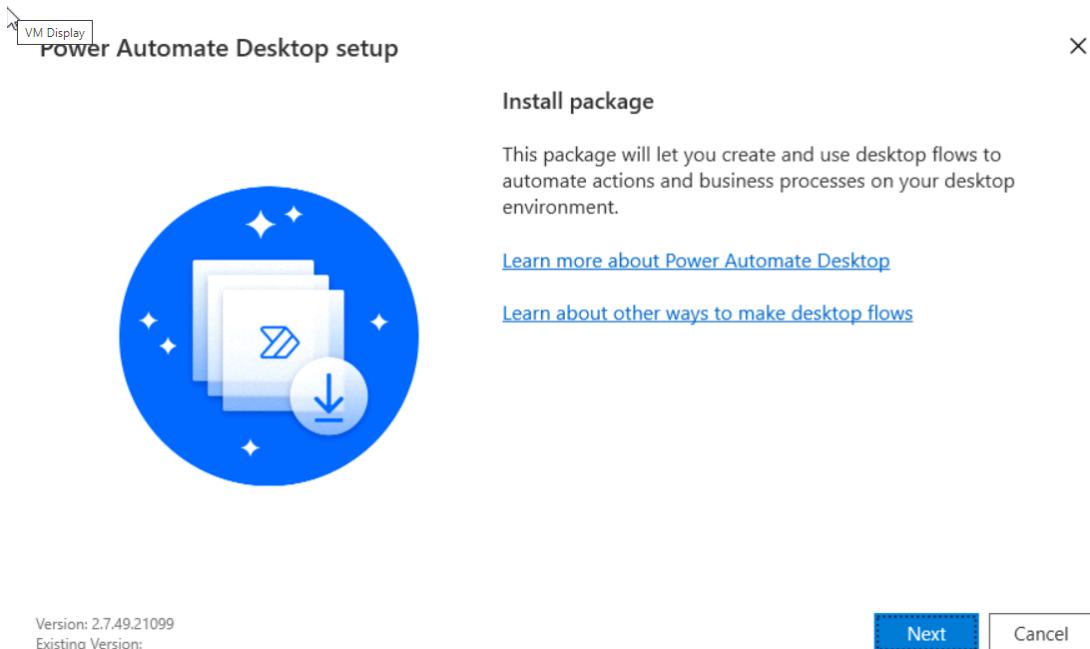
1. Go to **My flows** and click **Install**

The screenshot shows the Microsoft Power Automate "Flows" page. On the left, a sidebar menu includes "Home", "Action items", "My flows" (which is selected and highlighted with a red box), "Create", "Templates", "Connectors", and "Data". The main area has tabs for "Cloud flows", "Desktop flows", "Business process flows", and "Shared with me". A search bar at the top says "Search for helpful resources". On the right, there's a "Power Automate Desktop" section with a download button labeled "Install" (also highlighted with a red box) and a "Cloud flows" section with a download button labeled "Install". Below these sections, a message says "You don't have any flows" and "Choose from 100+ of templates to start automating your workflows."

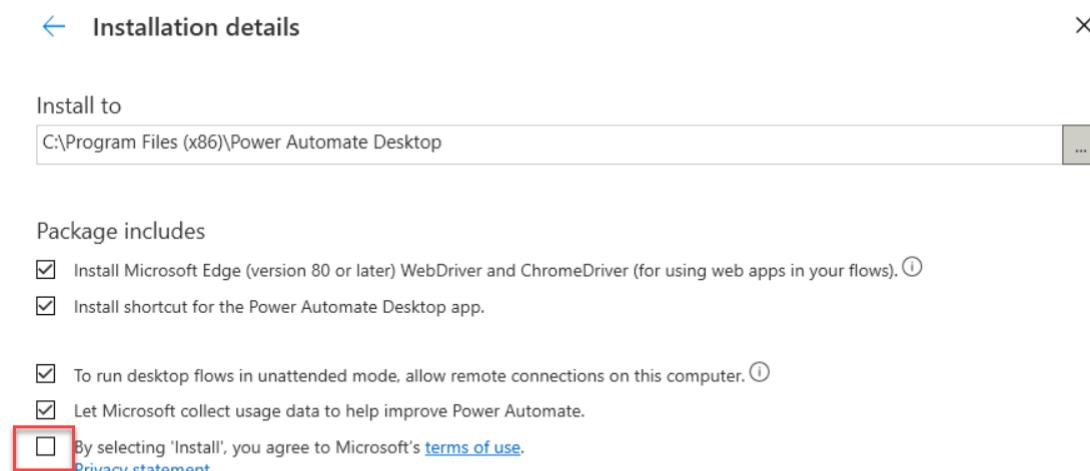
1. Click **Power Automate Desktop**
2. Click **Install** again
3. Click **On-premises data gateway**
4. Once the download completes, click on the file to open and run the installer:



1. On the following screen, click **Next**

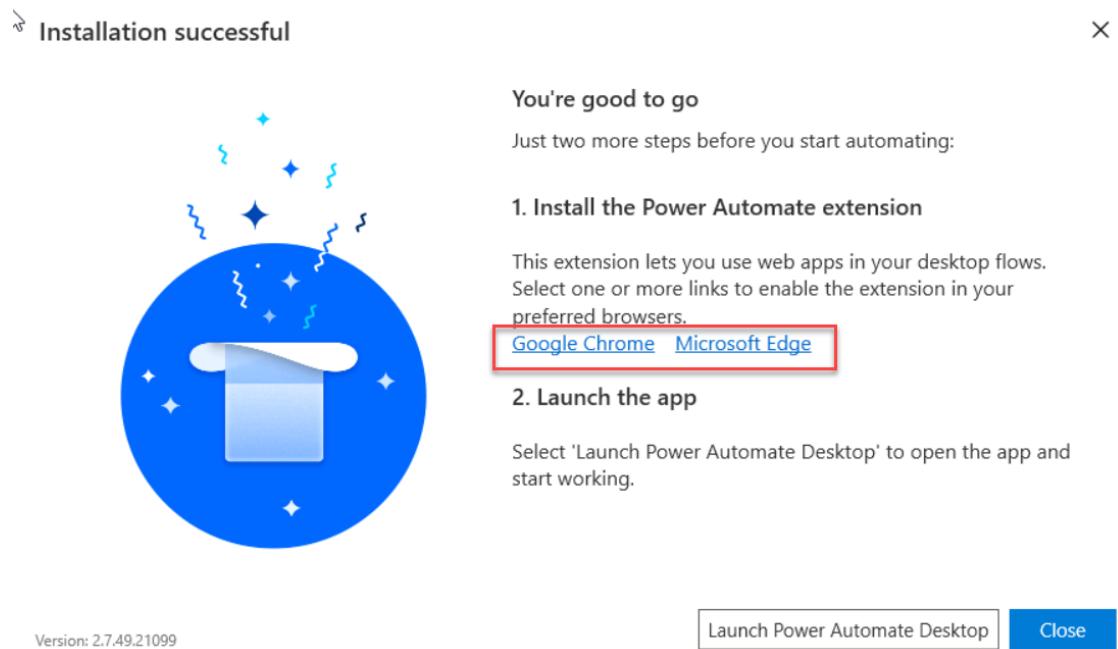


1. On the following screen, check the final box and click **Install**



Note: If prompted, click Yes

1. Once the installation completes, on the following screen, click on the link for **Google Chrome**



1. The Chrome web store will open as shown on the following screen

The screenshot shows the Microsoft Power Automate extension page in the Chrome Web Store. It features the Microsoft logo, the extension name "Microsoft Power Automate", and a "Add to Chrome" button. Below the extension details, it shows a rating of 4.5 stars from 16 reviews, categorized as "Productivity", and 100,000+ users.

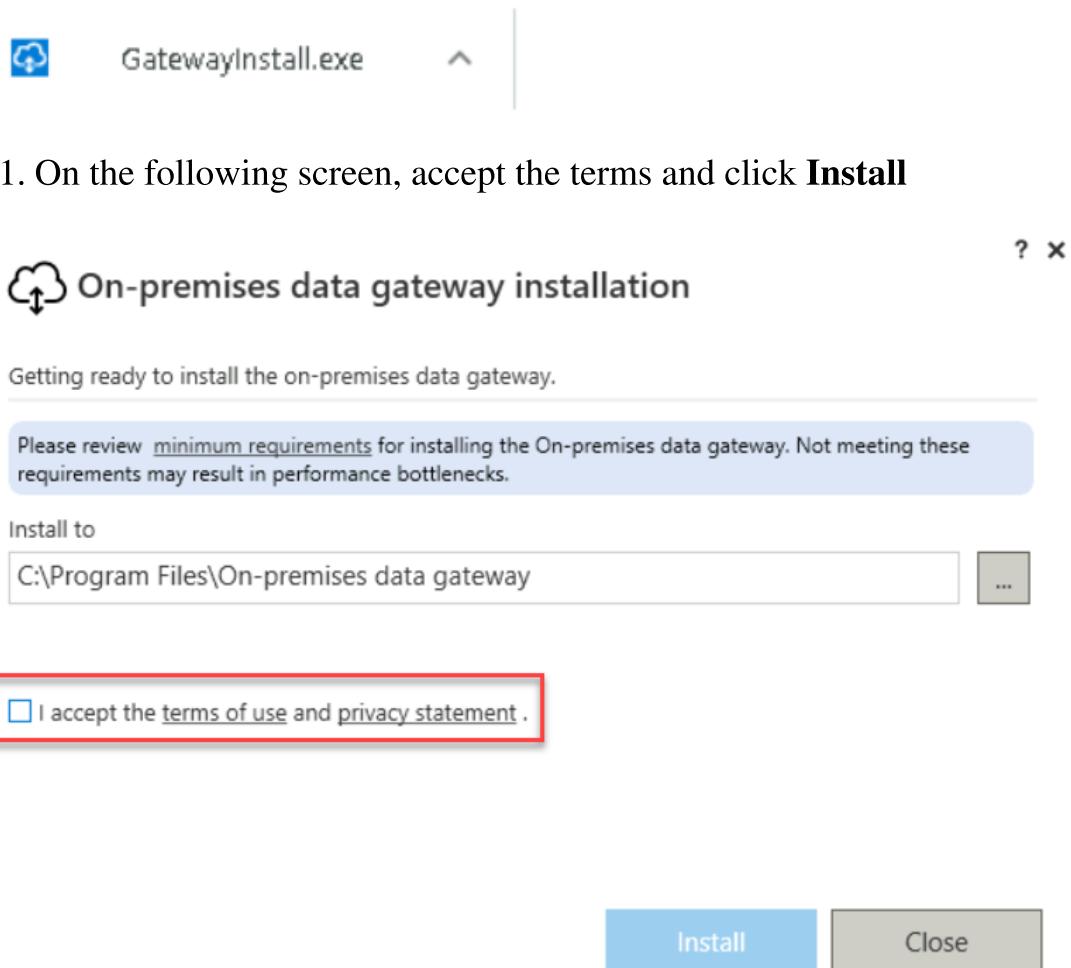
1. Click **Add to chrome** and **Add extension**
2. On the Power Automate Desktop setup screen, click on the link for **Microsoft Edge**
3. The Edge Add-ons page will open as shown on the following screen

The screenshot shows the Microsoft Power Automate extension page in the Microsoft Edge Add-ons store. It displays the same extension details as the Chrome store: "Microsoft Power Automate" by Microsoft Corporation, 4.5 stars from 17 reviews, and "Compatible with your browser". A "Get" button is visible on the right.

1. Click **Get** and **Add extension**
2. On the Power Automate Desktop setup screen, click **Close**

Task 3: On-premises data gateway

1. In the Power Automate portal, go to **My flows** and click **Install**
2. Click **Install**
3. Click **On-premises data gateway**
4. Once the download completes, click on the file to open and run the installer:



Note: If prompted, click Yes

1. Once the installation completes, enter your tenant email address and click **Sign in**



n-premises data gateway

? >

Almost done.

Installation was successful!

Email address to use with this gateway:

Next, you need to sign in to register your gateway.

Sign in

Cancel

1. On the following screen, register a new gateway and click **Next**

? X



On-premises data gateway

You are signed in as admin@WWLLAB015.onmicrosoft.com and are ready to register the gateway.

- Register a new gateway on this computer.
 - Migrate, restore, or takeover an existing gateway.
 - Move a gateway to a new computer
 - Recover a damaged gateway
 - Take ownership of a gateway
- The old gateway will be disconnected.

Next

Cancel

1. On the following screen, enter your name as the gateway name, enter **password** for the key, and click **Configure**



On-premises data gateway

You are signed in as admin@WWLLAB015.onmicrosoft.com and are ready to register the gateway.

New on-premises data gateway name

Add to an existing gateway cluster [Learn more](#)

Recovery key (8 character minimum)

(i) This key is needed to restore the gateway and can't be changed. Record it in a safe place.

Confirm recovery key

We'll use this region to connect the gateway to cloud services: West US 2 [Change Region](#)

[Provide relay details](#) By default, Azure Relays are automatically provisioned

Configure

Cancel

1. On the following screen, click **Close**

The screenshot shows the 'On-premises data gateway' status page. On the left is a sidebar with links: Status, Service Settings, Diagnostics, Network, Connectors, and Recovery Keys. The main area has a green checkmark indicating the gateway Instructor is online and ready. It shows the gateway version number as 3000.77.7 (March 2021 (Release 2)). There is a checkbox for sending usage information to Microsoft, which is checked, with a link to the privacy statement. Below this are three sections: 'Logic Apps, Azure Analysis Services' (West US 2), 'Power Apps, Power Automate' (West US 2) marked as 'Ready', and 'Power BI' (Default environment) marked as 'Ready'. At the bottom right is a 'Close' button.

1. In the Power Automate portal, go to **Gateways** to verify the gateway is configured

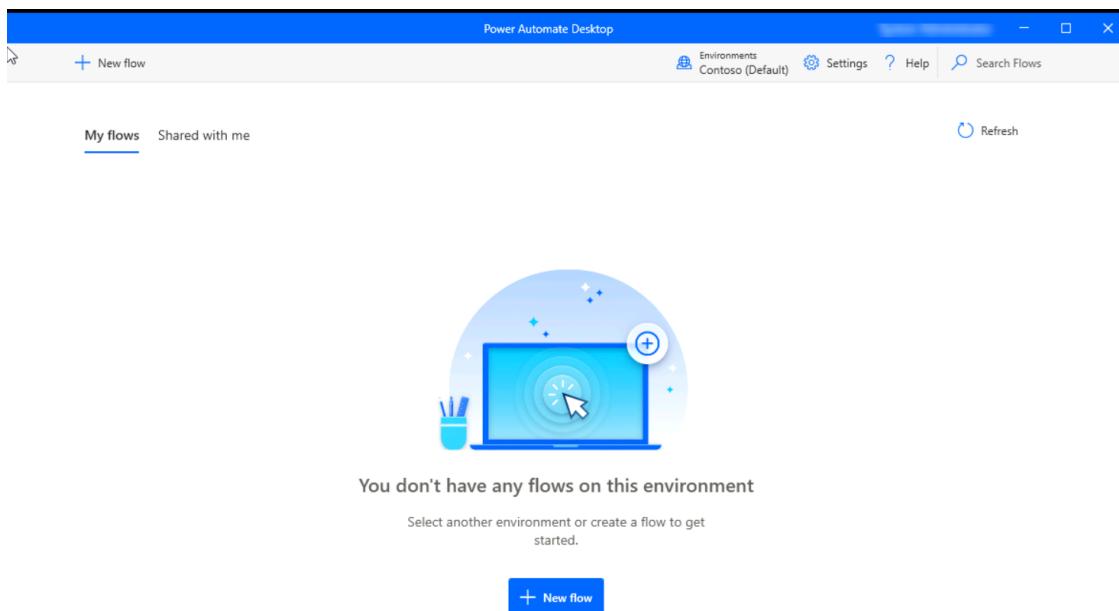
The screenshot shows the 'Power Automate' portal. The left sidebar includes Home, Action items, My flows, Create, Templates, Connectors, Data (Tables, Connections, Custom connectors), and Gateways. The 'Gateways' item is highlighted with a red box. The main area shows a 'Gateways' section with a table. The table has columns for Name and Contact information. One row is visible, showing a blurred name and contact information ending in '@onmicrosoft.com'. Below the table are links for 'Learn about gateways' and 'Troubleshoot a missing gateway'.

Exercise 2: Desktop flow

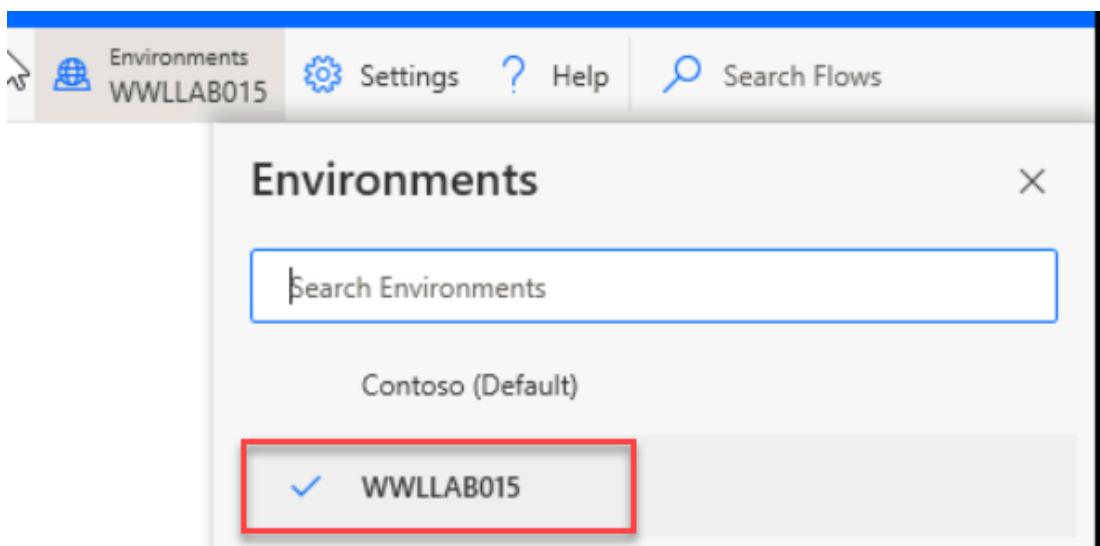
In this exercise you will create a desktop flow to enter an Invoice into the legacy Contoso Invoicing application.

Task 1: Capture tasks

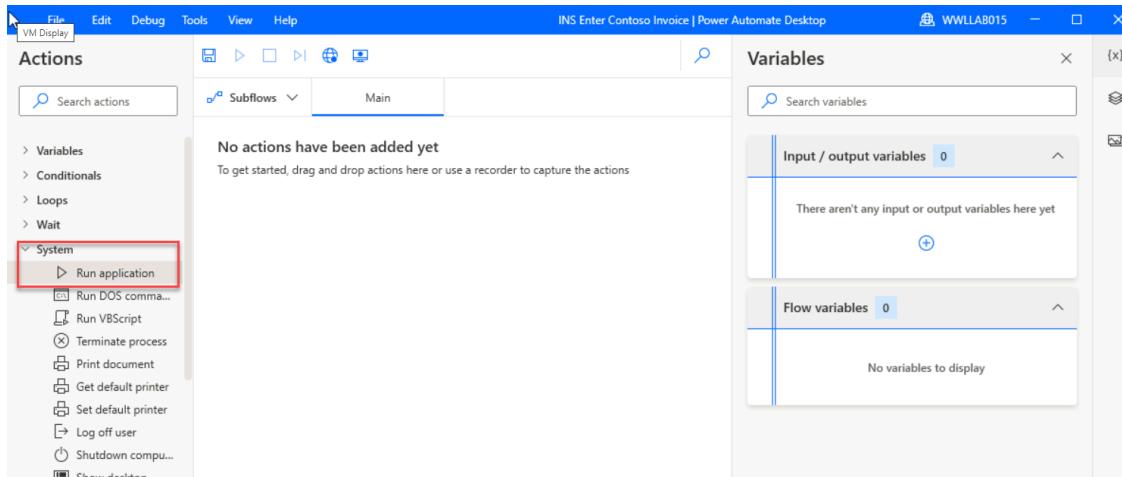
1. Launch Power Automate Desktop from the computer
2. Click **Sign in**
3. Provide your tenant credentials and click **Sign in**.
4. Provide your password and click **Sign in**



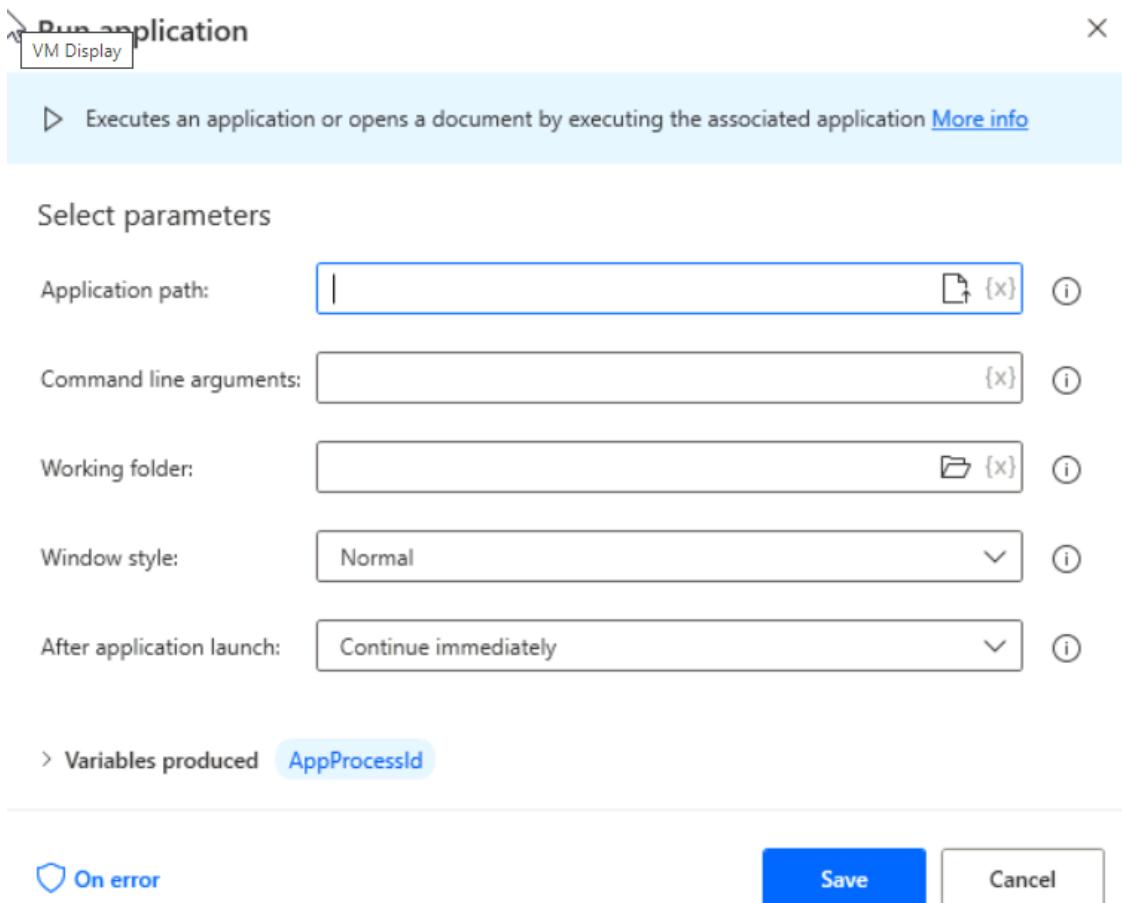
1. Select the production environment



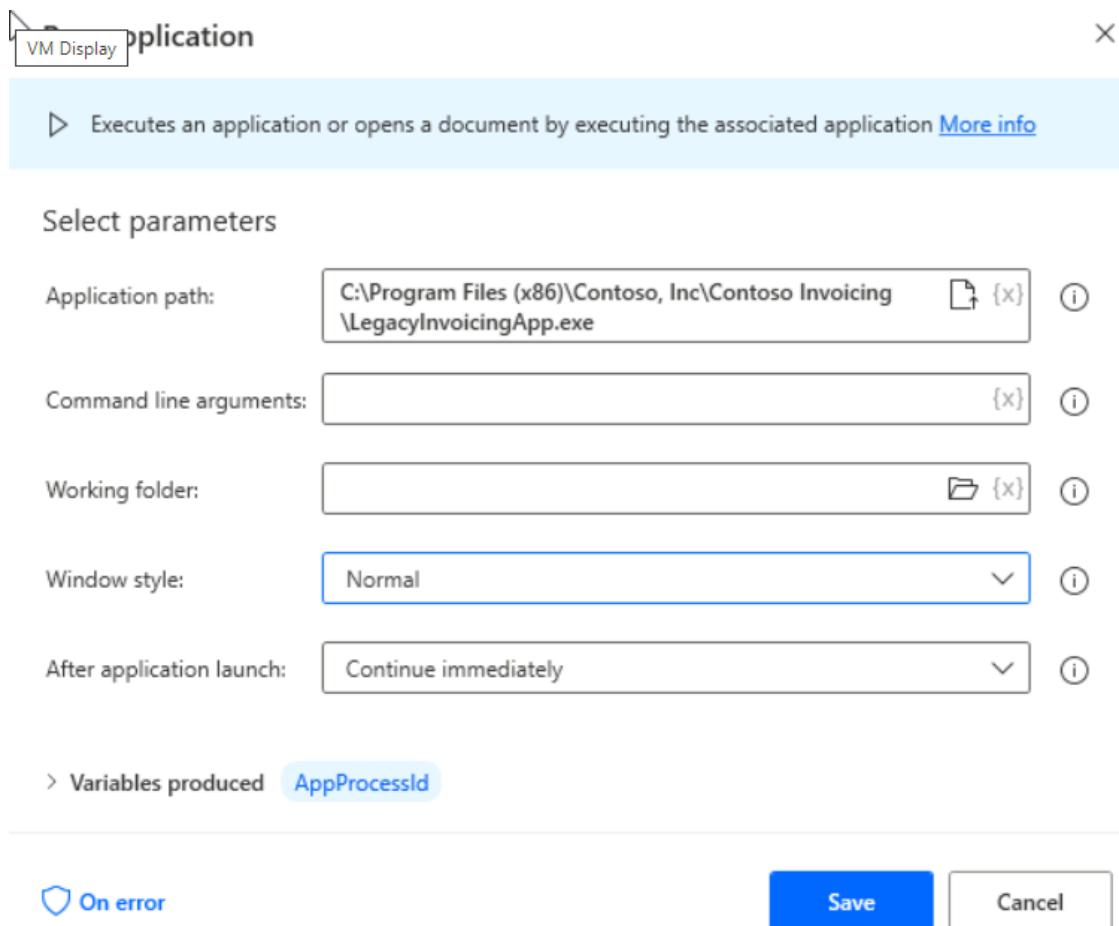
1. Click + New flow
2. Enter **Enter Contoso Invoice** prefixed with your initials for the name and click **Create**
3. When the application starts, expand the System actions and select Run application as shown in the following screenshot:



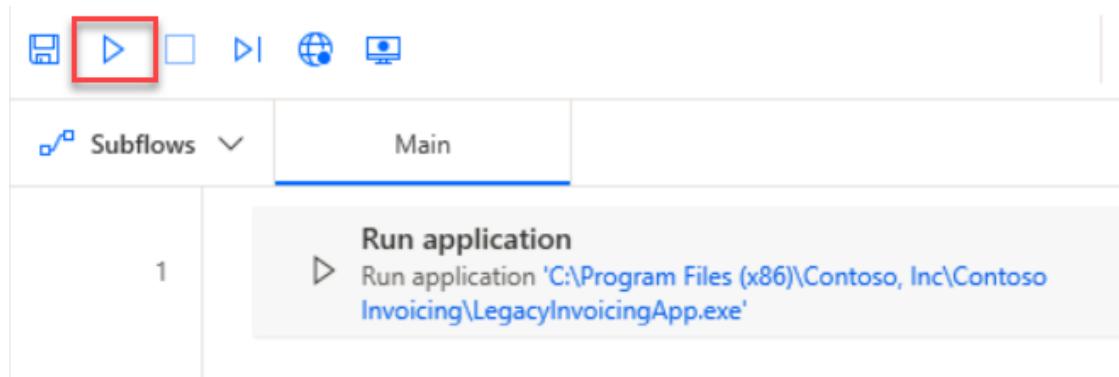
1. Drag the **Run** action to the canvas in the middle pane



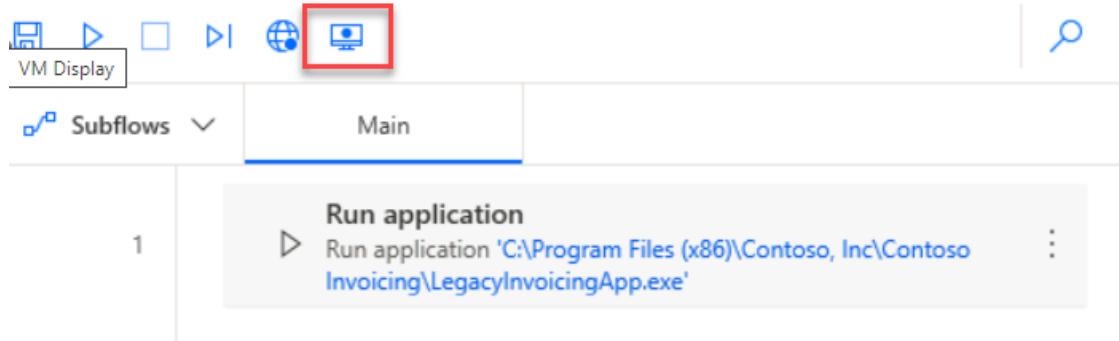
1. Enter the location path of Contoso Invoicing app under Application path and click **Save**



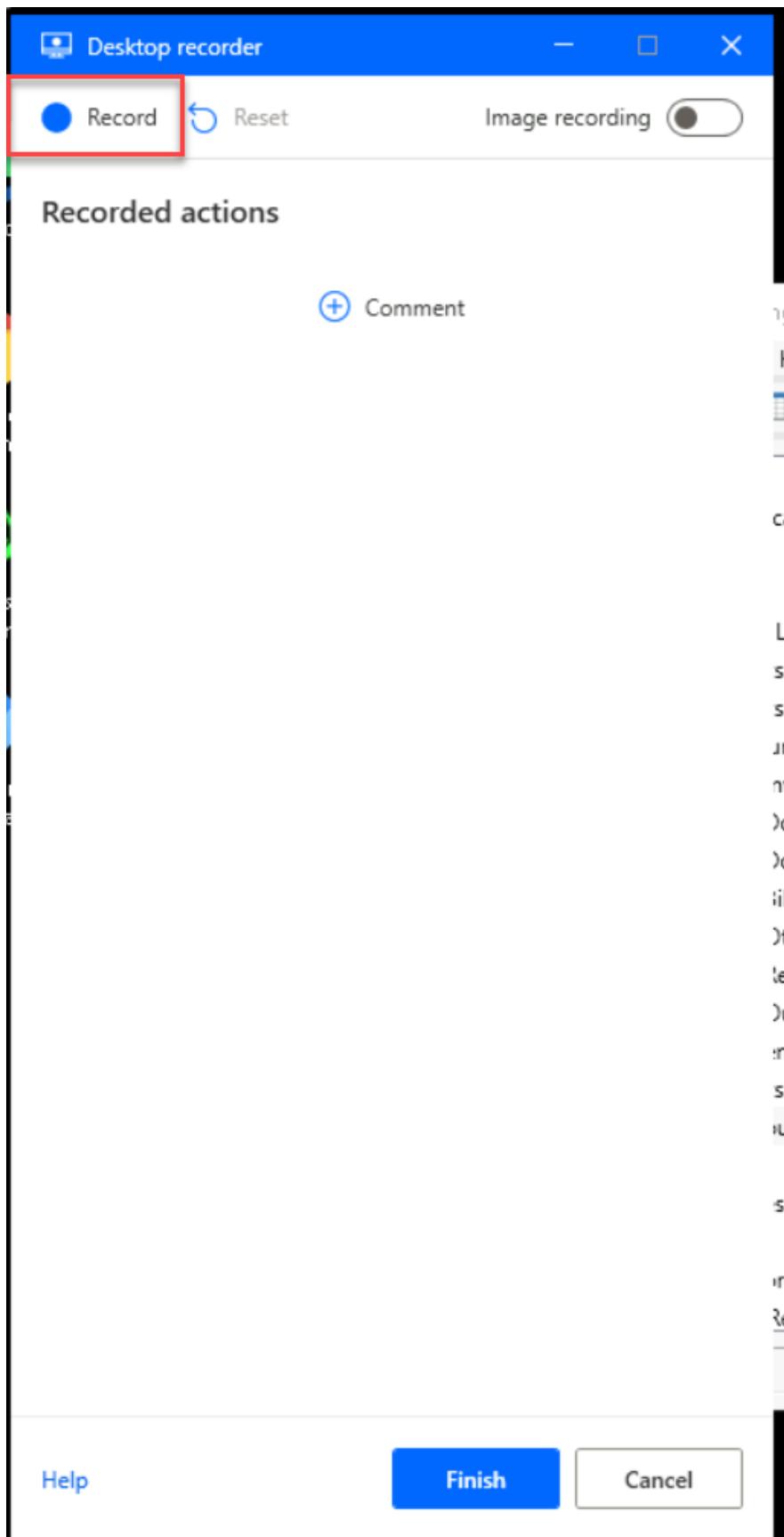
1. Click the **Run** icon to start the Contoso Invoicing app



1. Click the **Desktop recorder** icon to start the task recorder



1. Click the **Record** button to start recording steps



1. Click **Invoices** in the Contoso Invoicing app

The screenshot shows the Contoso Invoicing application interface. On the left is a navigation tree with categories like Office, Cross-Application Components, Logistics, Accounting (with sub-items General Ledger, Accounts Receivable, Accounts Payable, Text Entry, Documenting, Down Payment, Bill of Exchange, Other, Reference Documents, Outgoing Payment), Document, Accounts, Support, Cases, and Visitors. A red box highlights the 'Invoices' item under the 'Text Entry' section. To the right is a grid of account records:

ID	Account Name	Primary Contact	Contact Email
1001	Fabrikam	Invoicing Office	invoicing@fabrikam.com
1002	Litware Inc.	Andrew Dixon	adrew@litware.com
1003	Proseware, Inc.	Lynne Robbins	lrobbins@proseware.com
1004	Tailspin Toys	Pradeep Gupta	p.gupta@tailspintoy.com
1005	WingTip Toys	Bart Friday	b.friday@wingtip/toys.com

Below the grid, tabs for Account Detail, Alternates, Scheduling, and Agreements are visible. Under Account Detail, fields show ID: 1005, Account Name: WingTip Toys, Primary Contact: Bart Friday, and Contact Email: b.friday@wingtip/toys.com.

1. Click the **New Record** button in the Contoso Invoicing app

The screenshot shows the Contoso Invoicing application interface with the 'File' menu open. The 'New Record' button is highlighted with a red box and an arrow pointing to it. The menu bar also includes Edit, Notes, Help. Below the menu is a toolbar with icons for New Record, Open, Save, Delete, Find, and Print. A sidebar on the left shows a tree view with 'Cross-Application Comp' expanded.

1. In the Invoice Detail pane leave the date and enter the following:
 - a. Account: **This Account**
 - b. Contact: **This Contact**
 - c. Amount: **999**
 - d. Status: **Invoiced**

Invoice Detail [Notes | Budgeting | Sales | Resources]

ID: 1026

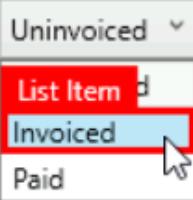
Date: 4/20/2021

Account: This Account

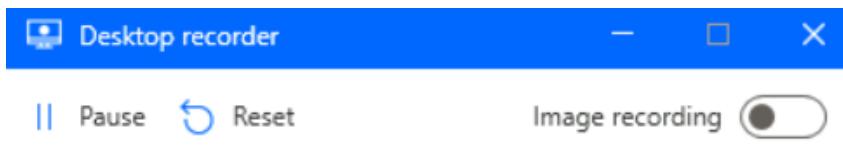
Contact: This Contact

Amount: \$999.00

Status



1. Click the **Save** button in the Contoso Invoicing app
2. Click the **Finish** button to stop recording steps



Recorded actions

Populate text field in window



Populate Edit 'TextBox' 3 on screen:

Window 'Contoso Invoicing' with Aa Text 999

Select tab in window



Select tab on UI element: Window 'Contoso Invoicing' on

screen: Tab Item 'Resources'

Click element in window



Left click on UI element: Combo Box 'ComboBox' on screen:

Window 'Contoso Invoicing'

Send keys in window



Send keys: {LMenu}{}{LMenu}

Click element in window



Left click on UI element: Text 'Invoiced' on screen:

Window 'Contoso Invoicing'

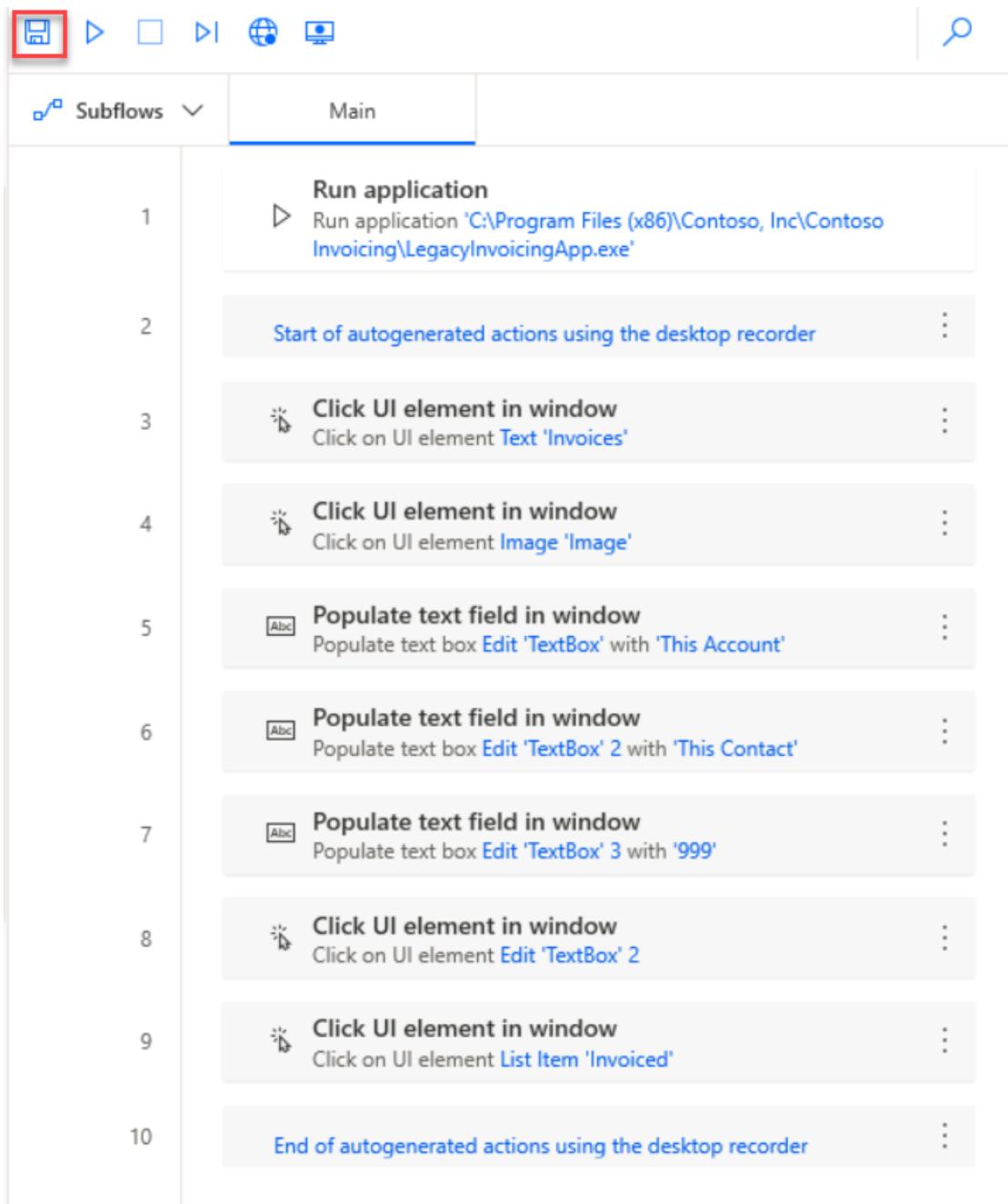
Comment

Help

Finish

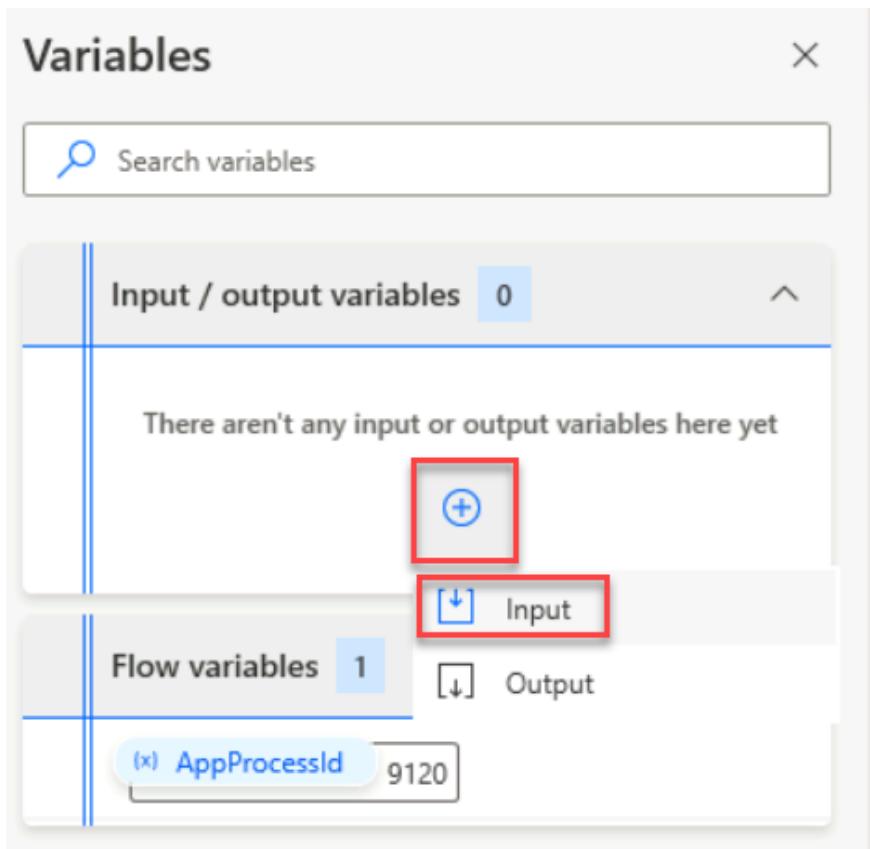
Cancel

1. Save the desktop flow



Task 2: Edit tasks

1. Click the + icon in the Input/Output variables pane and select **Input**



1. In the input variable dialog, enter the following and click **Create**:
 - a. Variable name: **AccountName**
 - b. Data Type: **Text**
 - c. DefaultValue: **TestAccount**
 - d. External name: **AccountName**
 - e. Description: **Name of account**

Add a new input variable

X

[+] Add a new variable to be used as input or output [More info](#)

Variable type:	<input type="text" value="Input"/> ⓘ
Variable name:	<input type="text" value="AccountName"/> ⓘ
Data type:	<input type="text" value="Text"/> ⓘ
Default value:	<input type="text" value="TestAccount"/> ⓘ
External name:	<input type="text" value="AccountName"/> ⓘ
Description:	<input type="text" value="Add an input description"/> ⓘ



Create

Cancel

1. Click the + icon in the Input/Output variables pane and select **Input**
2. In the input variable dialog, enter the following and click **Create**:
 - a. Variable name: **ContactName**
 - b. Data Type: **Text**
 - c. DefaultValue: **Your name**
 - d. External name: **ContactName**
 - e. Description: **Name of contact**

VM Display

Variable type:	<input type="text" value="Input"/>	(i)
Variable name:	<input type="text" value="ContactName"/>	(i)
Data type:	<input type="text" value="Text"/>	(i)
Default value:	<input type="text" value=""/>	(i)
External name:	<input type="text" value="ContactName"/>	(i)
Description:	<input type="text" value="Name of contact"/>	(i)

1. Click the + icon in the Input/Output variables pane and select **Input**
2. In the input variable dialog, enter the following and click **Create**:
 - a. Variable name: **Amount**
 - b. Data Type: **Text**
 - c. DefaultValue: **999**
 - d. External name: **Amount**
 - e. Description: **Invoice amount**

Variable type:	<input type="text" value="Input"/>	(i)
Variable name:	<input type="text" value="Amount"/>	(i)
Data type:	<input type="text" value="Text"/>	(i)
Default value:	<input type="text" value="999"/>	(i)
External name:	<input type="text" value="Amount"/>	(i)
Description:	<input type="text" value="Invoice amount"/>	(i)

1. Click the + icon in the Input/Output variables pane and select **Output**

2. In the input variable dialog, enter the following and click **Create**:

a. Variable name: **InvoiceReference**

b. Data Type: **Text**

d. External name: **InvoiceReference**

e. Description: **Invoice reference**

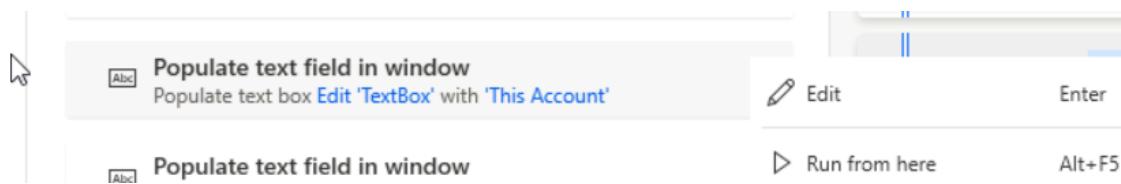
Add a new output variable X

Add a new variable to be used as input or output [More info](#)

Variable type:	Output	(i)
Variable name:	InvoiceReference	(i)
External name:	InvoiceReference	(i)
Description:	Invoice reference	(i)

Create Cancel

1. Edit the Account entry step



1. In the populate text field dialog, remove "This Account" and click on the {x} icon and select the **AccountName** variable

Populate text field in window

2 Fills a text box in a window with the specified text [More info](#)

3 Select parameters

4 Text box: %appmask['Window \'Contoso Invoicing\'']['Edit \'TextBox\'']% (i)

5 Text to fill-in: {x} (i)

Search

Input / output variables

Name	Type
> AccountName	Text value
> Amount	Text value

1. Click Save

Populate text field in window

VM Display

Fills a text box in a window with the specified text [More info](#)

Select parameters

Text box: %appmask['Window \'Contoso Invoicing\'']['Edit \'TextBox\'']% (i)

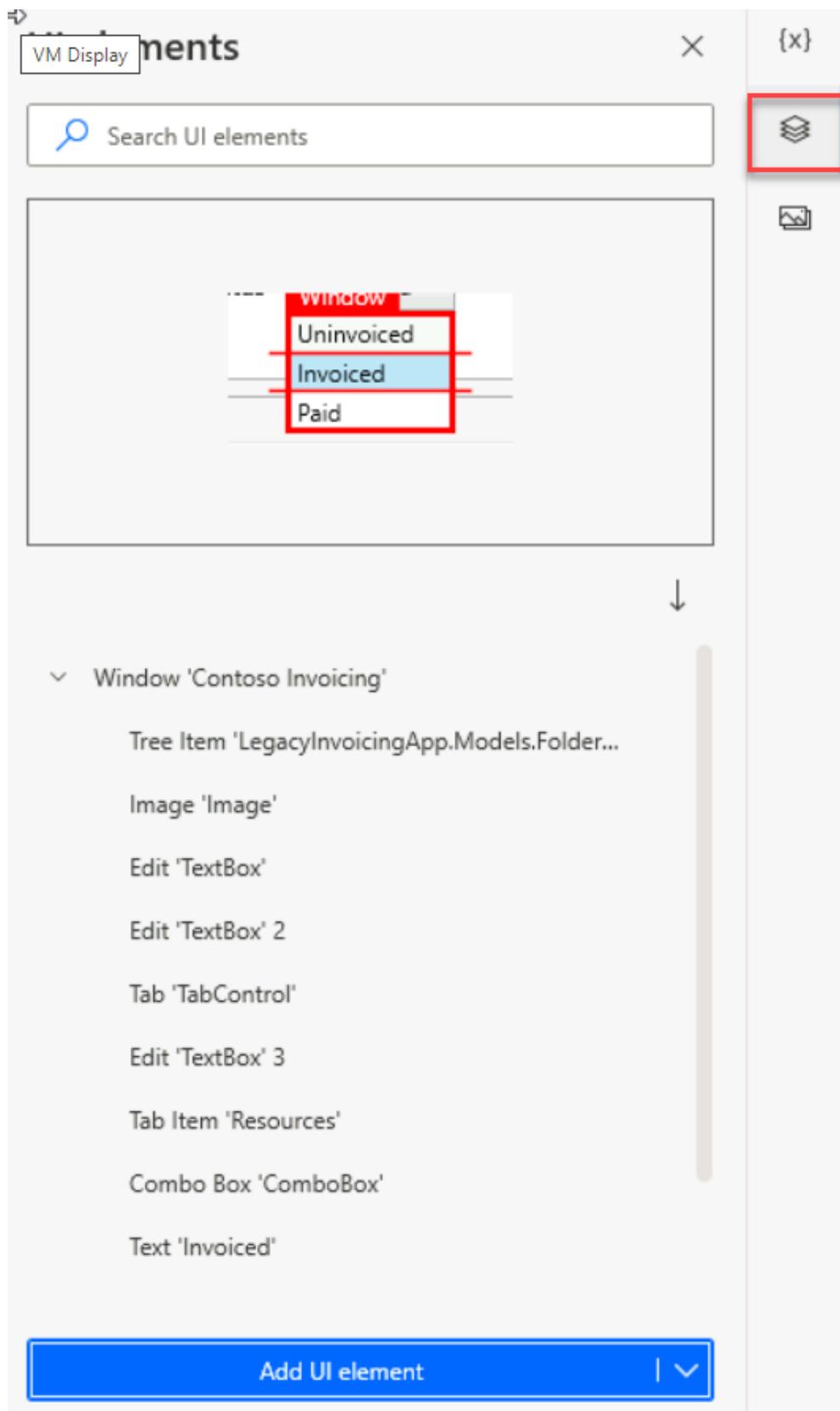
Text to fill-in: {x} (i)

%AccountName%

On error

Save Cancel

1. Edit the Contact entry step and replace the text with the **ContactName** variable
2. Edit the Amount entry step and replace the text with the **Amount** variable
3. Select the **UI elements** tab



1. Click **Add UI element**
2. Hover the mouse over the Invoice ID field in the Contoso Invoicing app

Invoice Detail	Notes	Budgeting	Sales	Resources
ID:	1026			
Date:	4/20/2021			
Account:	This Account			
Contact:	This Contact			
Amount:	\$999.00			
Status	Invoiced			

1. Hold the **Ctrl** key down and **Left-Click** on the ID field with the mouse
2. Click **Done**
3. Drag the **Get details of a UI element in Window** action to the end of the steps in the canvas, and select the Invoice ID UI element

2 Get details of a UI element in window

3 Gets the value of a UI element's attribute in a window [More info](#)

4 Select parameters

UI element:

UI elements 13 

- Window 'Contoso Invoicing'
- Combo Box 'ComboBox'
- Edit 'TextBox'
- Edit 'TextBox' 2
- Edit 'TextBox' 3
- Image 'Image'
- List Item 'Invoiced'
- Tab Item 'Resources'
- Tab 'TabControl'
- Text 'Invoiced'
- Text 'Invoices'
- Tree Item 'LegacyInvoicingApp.Models.FolderModel'
- Text '1026'**

Detail Notes E

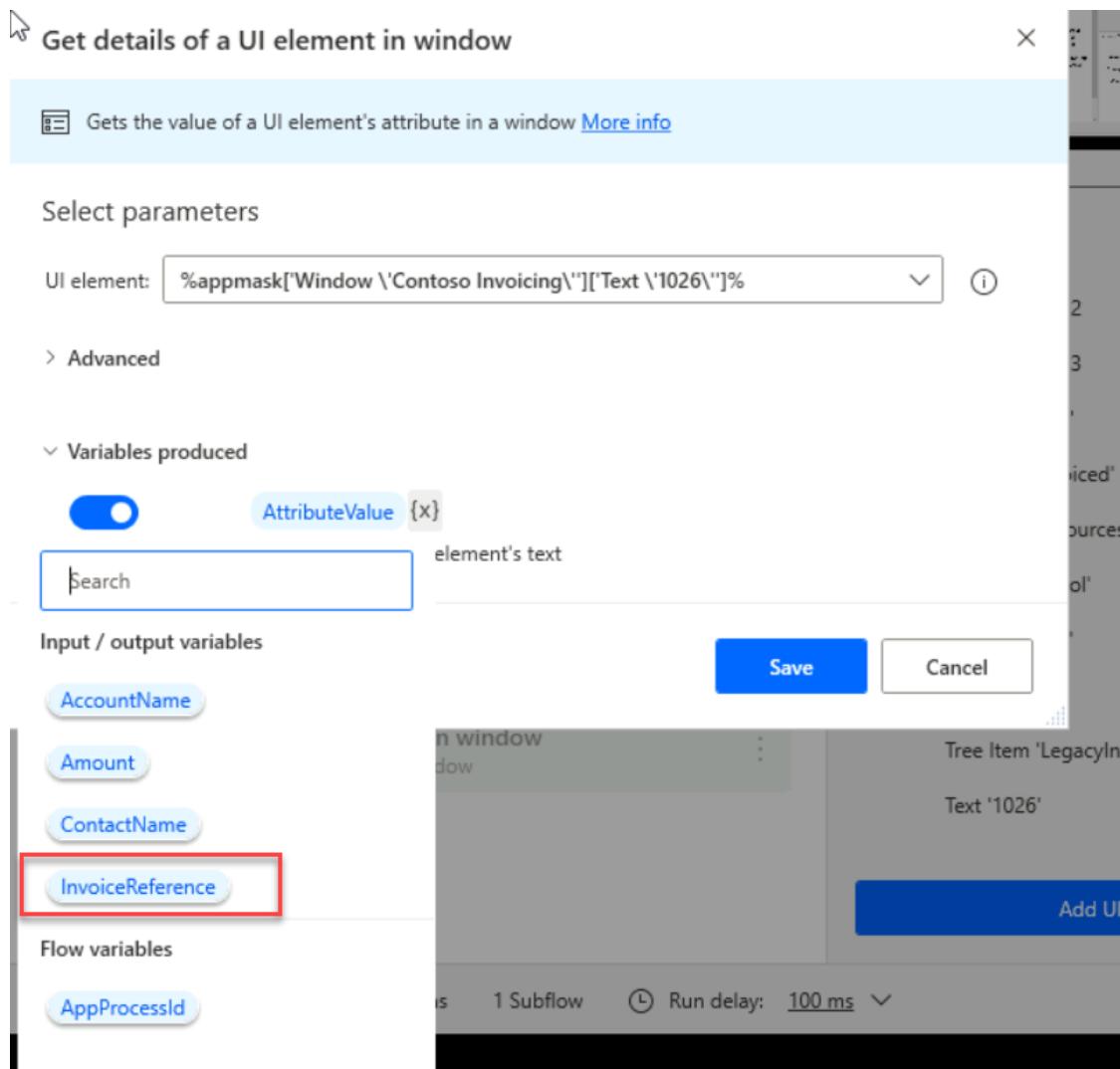
ID: 1026

Date: 4/20/2021

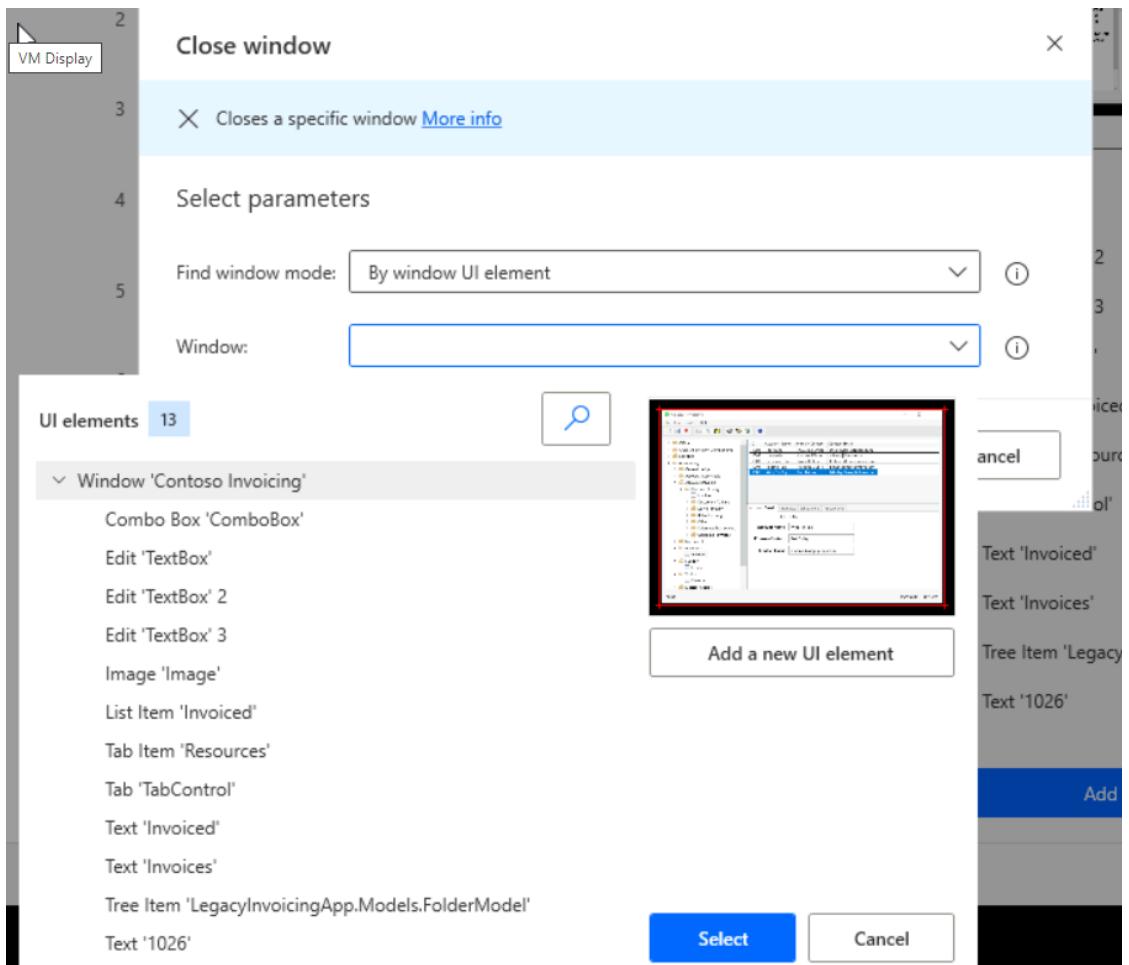
Add a new UI element

Select **Cancel**

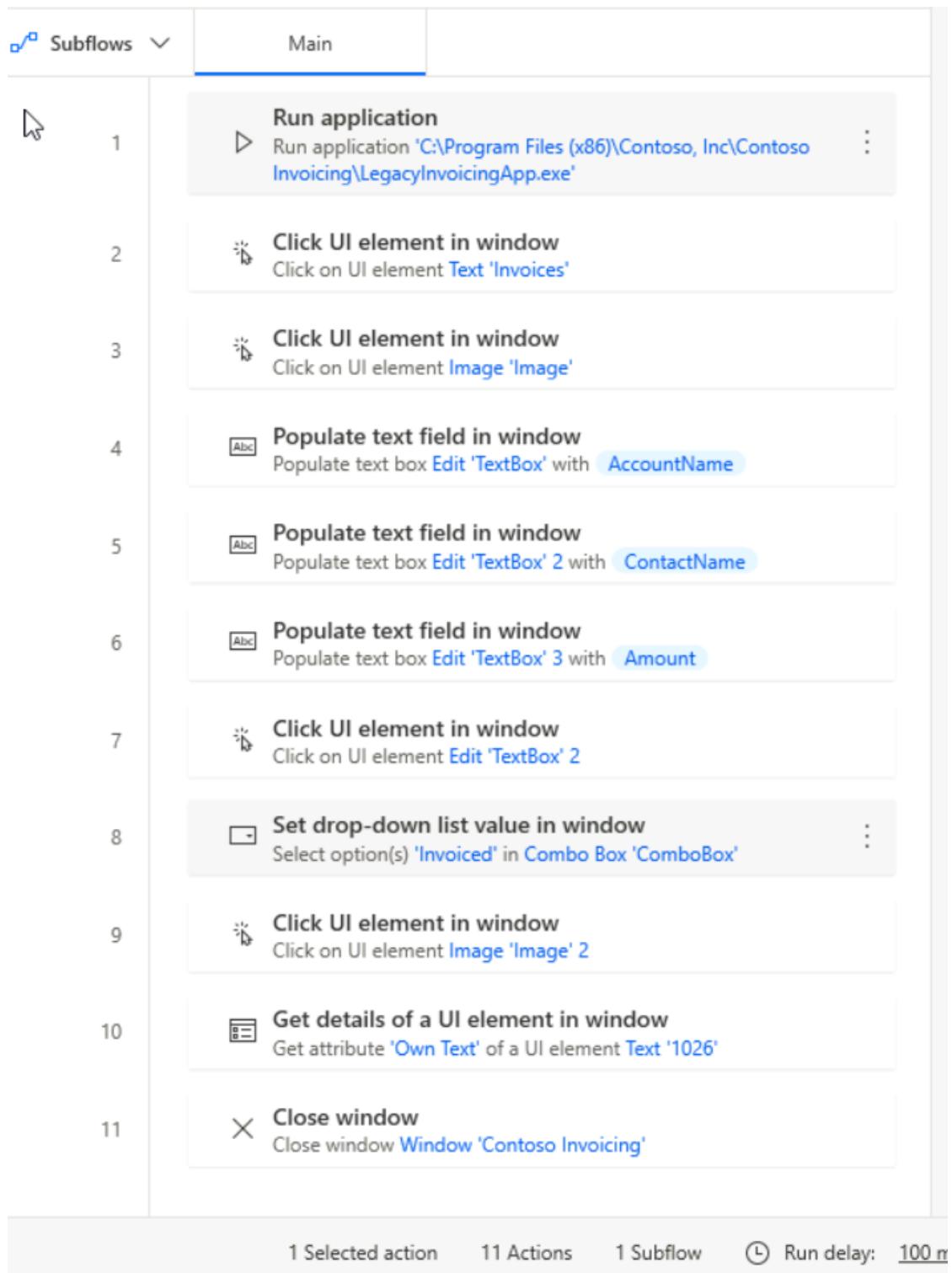
1. Set the variable produced to be the output variable and click **Save**



1. Drag the **Close window** action to the end of the steps in the canvas, and select the Contoso Invoicing UI element



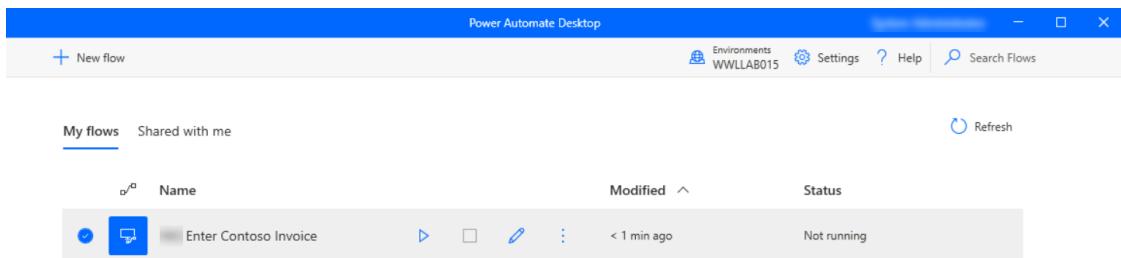
1. **Save** the desktop flow. The desktop flow steps should be similar to the following screenshot



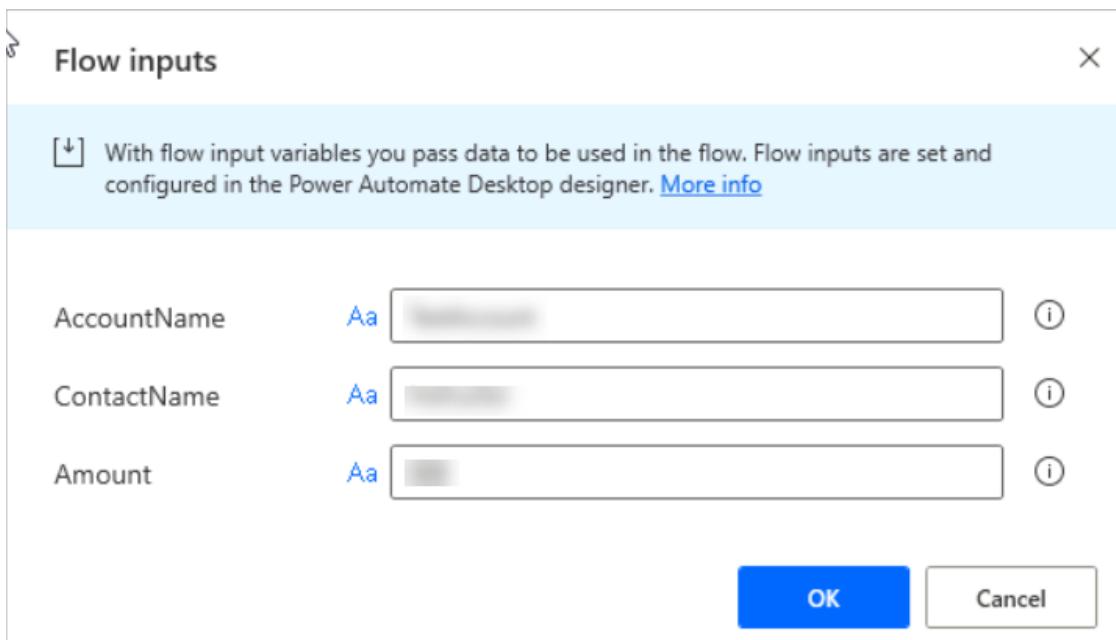
1. **Close** the Power Automate Desktop editor window

Task 3: Test

1. From the Power Automate Desktop console, run the desktop flow

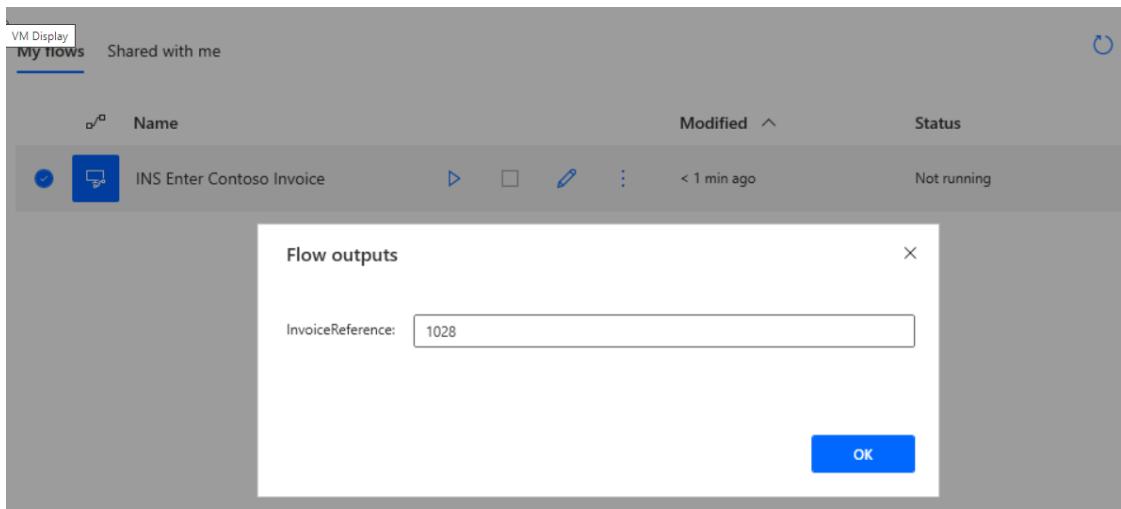


1. Enter the inputs for the flow



1. Monitor the running of the flow in attended mode

2. Verify a new Invoice reference is output

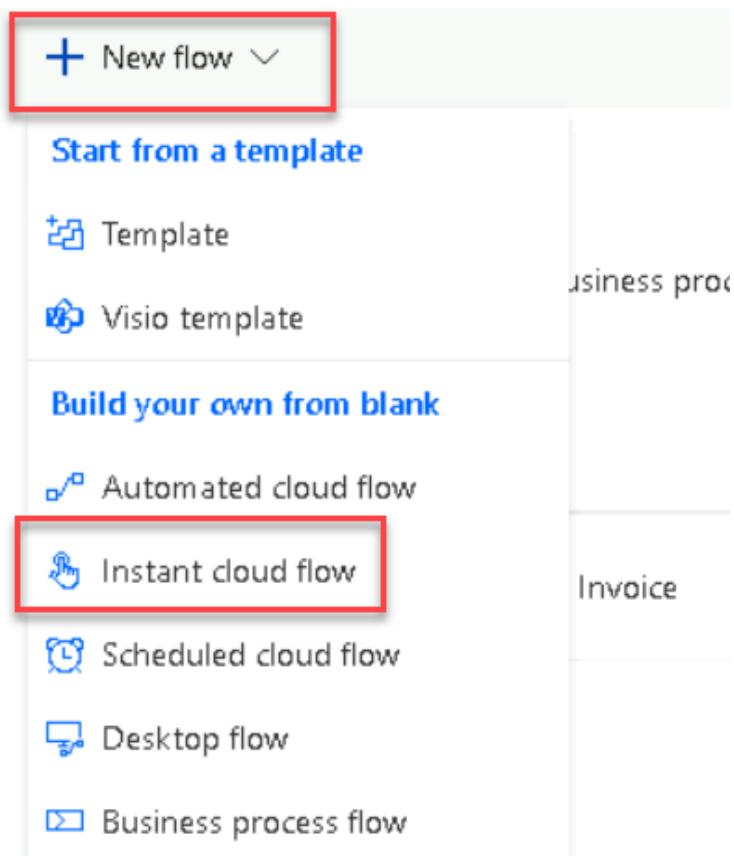


Exercise 3: Cloud flow

In this exercise you will create and run a cloud flow to retrieve data from Dataverse and use the desktop flow to enter the data into the legacy Contoso Invoicing application.

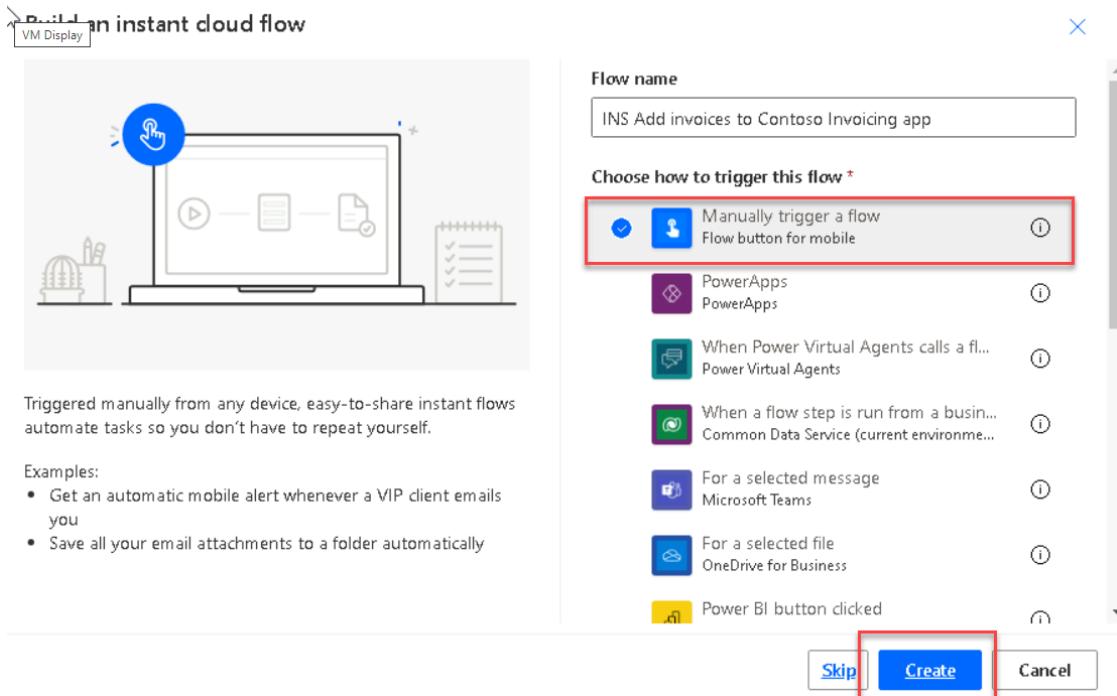
Task 1: Create Button flow

1. Navigate to <https://flow.microsoft.com>
2. Click **Sign in**
3. Provide your tenant credentials and click **Next**.
4. Provide your password and click **Sign in**
5. Select the production environment
6. Go to **My flows** and select the **Desktop flows** tab
7. Verify the desktop flow you created is listed
8. Click **+ New flow** and select **Instant cloud flow**



1. Name the flow **Add Dataverse invoices to Contoso Invoicing app** prefixed with your initials

2. Select **Manually trigger a flow** and click **Create**



1. Click **+ New step**

2. Search for and select the **Dataverse** connector

↓

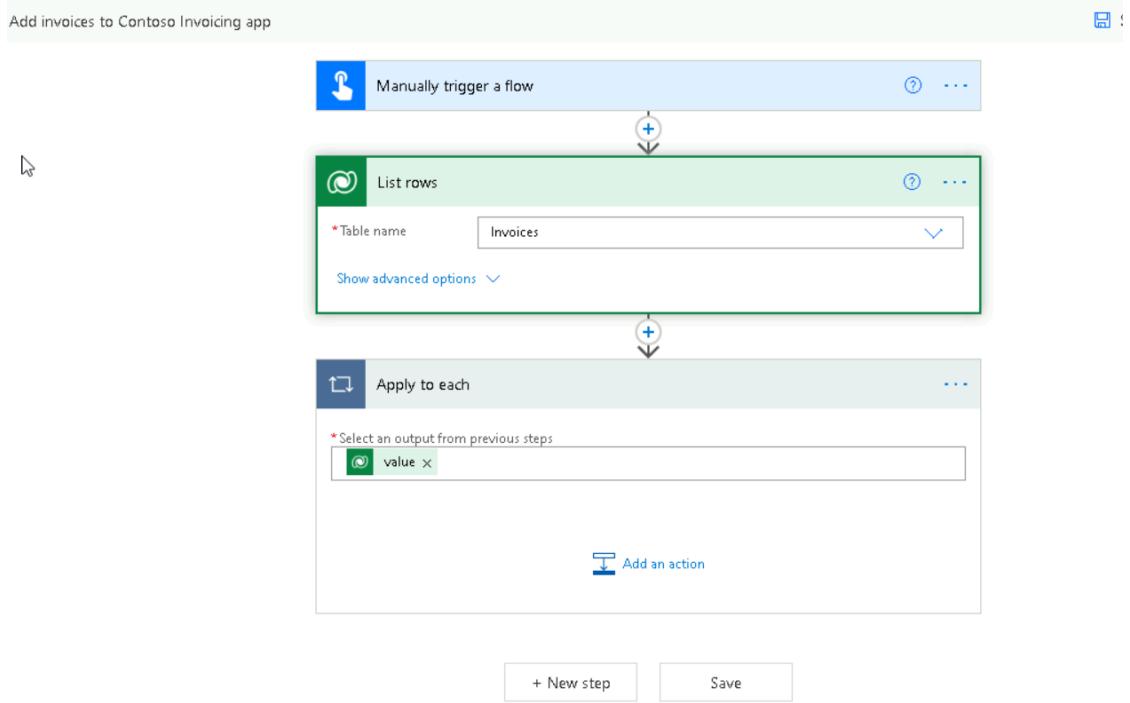
Microsoft Dataverse

Search connectors and actions

Triggers Actions See more

Action	Category	Status
Add a new row	Microsoft Dataverse	PREMIUM
Delete a row	Microsoft Dataverse	PREMIUM
Download a file or an image	Microsoft Dataverse	PREMIUM
Get a row by ID	Microsoft Dataverse	PREMIUM
List rows	Microsoft Dataverse	PREMIUM
Perform a bound action	Microsoft Dataverse	PREMIUM
Perform an unbound action	Microsoft Dataverse	PREMIUM
Predict using AI Builder models	Microsoft Dataverse	PREMIUM
Relate rows	Microsoft Dataverse	PREMIUM

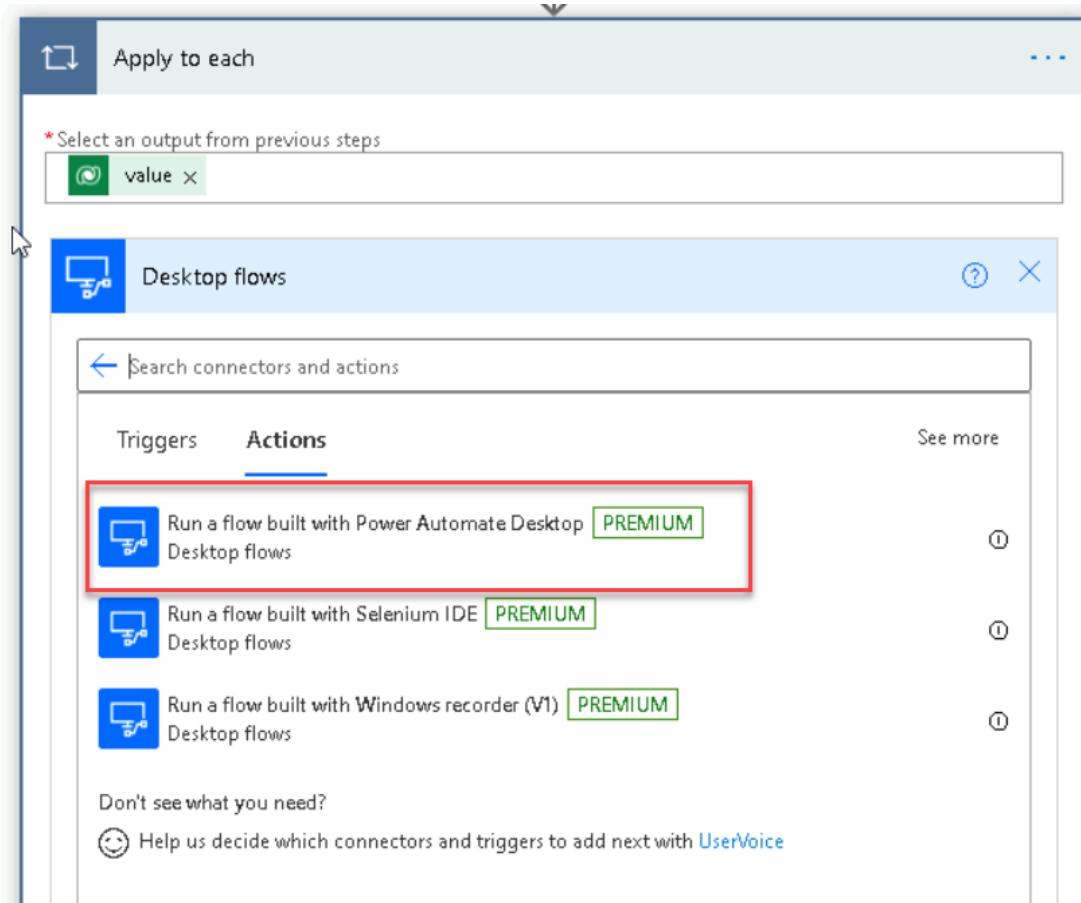
1. Select the **List rows** action
2. Select the **Invoices** table
3. Click **+ New step**
4. Select the **Control** connector and the **Apply to each** action
5. Select **value** from the List rows step in the Apply to each



1. Save the flow

Task 2: Call desktop flow

1. Navigate to <https://flow.microsoft.com>
2. Select the production environment
3. Go to **My flows** and select the **Cloud flows** tab
4. Edit the **Add Dataverse invoices to Contoso Invoicing app** flow
5. Expand the **Apply to each** step
6. Click **Add an action**
7. Search for **Desktop** and select the **Desktop flows** connector



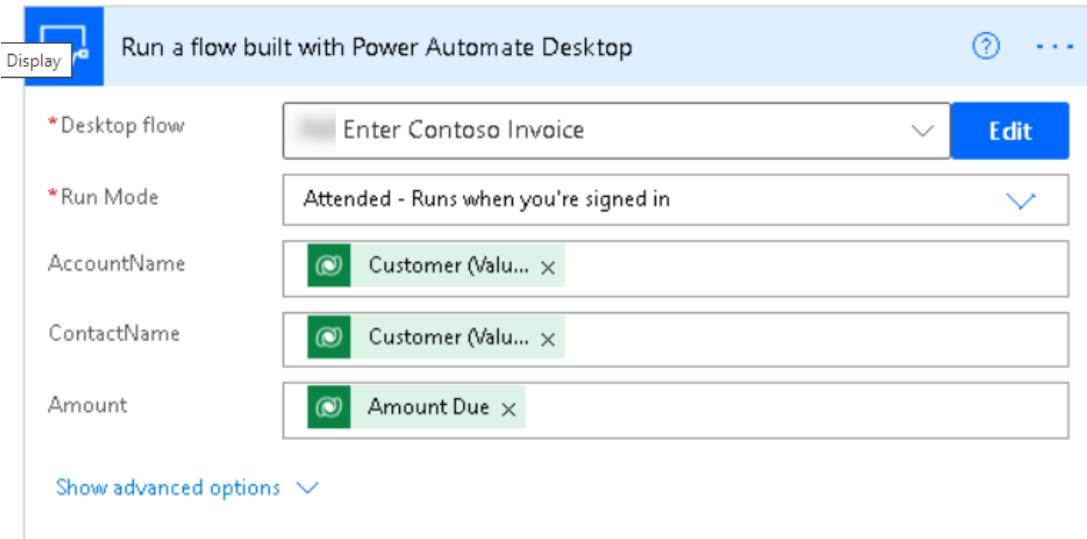
1. Select **Run a flow built with Power Automate Desktop** and click **Create**
2. Select the gateway configured earlier
3. Use the username and password for your computer, and click **Create**

The screenshot shows the 'Create gateway' dialog box. The title bar says 'Desktop flows'. The main area contains instructions: 'Use your credentials to create a connection to an on-premises data gateway and securely connect your locally stored data to Power Automate and other Microsoft services.' Below the instructions are three input fields with validation stars (*):

- * Gateway name: A dropdown menu with a placeholder 'Select gateway' and a 'New gateway' button.
- * Domain and username: An input field containing '@WVLLAB015.onmicrosoft.com'.
- * Password: A masked input field showing '*****'.

At the bottom right of the dialog is a large blue 'Create' button.

1. Select your desktop flow and choose **Attended** mode
2. Use Dynamics Content to select the Account and Contact as **Customer (value)** and Amount as **Amount Due**



1. Save the flow

Task 3: Run the flow

1. Navigate to <https://flow.microsoft.com>
2. Select the production environment
3. Go to My flows and select the Cloud flows tab
4. Run the Add Dataverse invoices to Contoso Invoicing app flow
5. Click Continue
6. Click Run flow
7. Click Done
8. View the run history for the flow and verify the flow ran successfully
9. Start the Contoso Invoicing app and check that the invoices were added successfully

Table of Contents

It is strongly recommended that MCTs and Partners access these materials and in turn, provide them separately to students. Pointing students directly to GitHub to access Lab steps as part of an ongoing class will require them to access yet another UI as part of the course, contributing to a confusing experience for the student. An explanation to the student regarding why they are receiving separate Lab instructions can highlight the nature of an always-changing cloud-based interface and platform. Microsoft Learning support for accessing files on GitHub and support for navigation of the GitHub site is limited to MCTs teaching this course only. ⁷

Download Latest Student Handbook and AllFiles Content	1
{ { activity.lab.title } }{ % if activity.lab.type % } - { { activity.lab.type } }{ % endif % }	1
{ { activity.demo.title } }	1