MB-600: Dynamics 365 + 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 /home/ll/Azure_clone/Azure_new/MB-600-Dynamics-365-Power-Platform-Solution-Architect/../../releases/latest**
- **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 MB600 student tenant
==========

This document is intended to help setup a tenant provisioned from https://cdx.transform.microsoft.com/ to run the workshop. The script provides a set of commands to automate creating users, assigning licenses and creating a trial environment with Common Data Service for each user.

Do not use this on your production live tenants as it does delete users and environments. You will create a tenant at demos.microsoft.com and make sure to use those credentials.

This setup process is documented to use the Windows PowerShell ISE application. If you aren���t familiar with it you can find more information here https://docs.microsoft.com/en-us/powershell/scripting/components/ise/introducing-the-windows-powershell-ise?view=powershell-6

Step 1 ??? Create a new Demo tenant

1. Sign up for a new demo at https://cdx.transform.microsoft.com/
2. Select **My Environments -> Create Tenant**
3. Select **Type of Quick Tenant**
4. Choose **Microsoft 365 Enterprise with Users and No Content.** Create the tenant.
5. Make note of the admin user and password. You might have to wait a minute or two for the tenant to be provisioned first before they are available.
6. **Open a new private browser session** and go to admin.microsoft.com. Login with your tenant credentials.
7. Go to **Billing ->Purchase Services** and Search on **Power Apps per User Plan** and click on it.
8. Click **Get free trial**.
9. Click **Try now**.
10. Click **Continue**.
11. Now assign one of these licenses to your admin user, this will leave 24 that you can use with the remaining script setup.
12. In the same browser sessions, navigate to aka.ms/ppac
13. Select **Environments** and make sure you can see the default environment.
14. Click on the default environment it will take you to admin.powerapps.com
15. Change the environment name to **Personal Productivity** and click **Save**.

Step 2 - Run Setup Script

Note: If you receive errors or messages not documented in these steps, review the known issues below for possible resolution.

Wait and give some time for the tenant to sync user licensing.

1. From Windows Start search on and run the Windows PowerShell ISE application.

2. File -> Open **Setup.ps1.**

3. Run the script by pressing F5 or by clicking the play icon. This just loads the setup commands it does not start the actual setup process. You will know this is completed when the command prompt becomes available again.

4. If you see errors or warnings resolve them prior to continuing

5. **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 this script which will remove all PowerShell modules used by the setup script so the setup script can then install the versions it requires.

6. Run the **Start-MB600-Setup** command; use the parameters to provide tenant and configuration information like shown in the following example commands:

    - North America tenant example:
      Start-MB600-Setup -TenantName MX12345ABC -CDSLocation unitedstates -UserCount 24

    - Europe tenant example:
      Start-MB600 -Setup -TenantName MX12345ABC -CDSLocation europe
      -UserCount 24

    - When choosing your User Count keep in mind the number of licenses you have available

- When you are prompted for your tenant admin account **make sure to provide the account only for your demo tenant admin.**

- You can get parameter help using **get-help Start-MB600-Setup** -Detailed

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.

It���s always good to manually review a few of the users and environments prior to your event.

The users that are created 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.

Other commands available

- In the event CDS provisioning does not complete you can run just that using the **Resume-MB600-CDSProvisioning** command to finish that part of setup.

Known Issues

- You might receive warning messages about PSGallery ��? if you wish to suppress those messages you can use the following command to change your trust level on the gallery

  ```
  Set-PSRepository -Name PSGallery -InstallationPolicy
  Trusted
  ```

- Some computers have restrictive policies for running unsigned scripts ��? you can adjust your execution policy using the information provided here if needed https:/go.microsoft.com/fwlink/?LinkID=135170

- Sometimes on copy and paste line spacing loses formatting. If the behavior is not as expected, review the spacing. Copyright ��

Content

Introduction to the Labs

Objectives

Purpose

The purpose of the hands-on labs is to reinforce learning.

Preparation

To execute the labs, you need access to a Power Platform environment where you have access to create apps.

Resources

The labs utilise a Microsoft 365 tenant with:

Power Apps: 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.

Implications of using a cloud service

The user interface and functionality in Microsoft 365 and Power Platform are subject to change without notice so the steps listed may not be correct.

Location of files

Files are in GitHub https://github.com/MicrosoftLearning/MB-600-Dynamics-365-Power-Platform-Solution-Architect/tree/master/Allfiles

Credentials

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:

1. Use InPrivate (Edge) or Incognito (Chrome)

2. 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 Setup

Trial tenant

Objectives

Create a Power Apps trial.

Steps

Sign up for a trial

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

! /home/ll/Azure_clone/Azure_new/MB-600-Dynamics-365-Power-Platform-Solution-Architect/Allfiles/media/fdfc45f38beb293323826d9f71469bb8.png

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.

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

Fill out the form above.

For Company Name, use **Power Platform Training**.

For Your organisation size, select **1 person.**

Note: We recommend using United States for country and English as language as this makes following the labs easier.

Click **Next**.

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

Enter the unique verification code you received and click **Next**.

Fill out the form above.

For the organisation name, enter a unique name to reflect this training e.g., your initials + year/month.

Click **Check availability**.

Click **Next**.

Fill out the form above.

Click **Sign up**.

After a short while the following page will appear.

Write the user ID down.

Click **Get Started**

Configure Microsoft 365

This will transfer control to the Microsoft 365 environment. You need to access the Microsoft 365 Admin Centre https://admin.microsoft.com/AdminPortal/Home.

Click **Users** and then **Active Users** in the left hand navigation

Verify that your users has all licenses assigned

Close the pane.

Lab Setup

Objectives

Create users and environments for all students.

Steps

PowerShell Script

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.

Change Directory to where the **Setup.ps1** and **ContosoDeviceOrderManagement_1_0_0_1.zip** files are located.

Run the following commands:

Set-ExecutionPolicy RemoteSigned

Set-PSRepository -Name PSGallery -InstallationPolicy Trusted

If prompted answer with **A [Yes to All]**

Run **Setup.ps1**

If prompted answer with **A [Yes to All]**

Run the following command:

Start-MB600-Setup -TenantName \<Tenant name> -CDSLocation \ <Region> -UserCount \<Number of Students>

Your Tenant name is the name you entered for Step 3 when creating the trial. You should enter only the first part of the Tenant name i.e., without onmicrosoft.com.

The region should be the region of your tenant and will be from the country you selected in Step 2. 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 script creates environments and databases for each student and a development environment with the solution imported.

Check Dataverse Environments

Navigate to the Power Platform Admin Center https://aka.ms/ppac

Verify that there are environments for each student.

Check Development Environment

Navigate to the Maker Portal https://make.powerapps.com/

Select the **Device Ordering Development** environment.

If you do not see the environment, press Ctrl-F5 to refresh your browser.

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

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

Add users to the Development environment

Navigate to the Power Platform Admin Center https://aka.ms/ppac

Select the **Device Ordering Development** environment.

Click on **Settings**.

Expand **Users + permissions**.

Select **Users**.

If the Lab Admin users for each of the students are listed, skip to the next step, Assign security roles.

Click on **+ Add user**.

Enter **labadmin**

Select a Lab Admin user and click **Add**.

Click **Not Now**.

Repeat for all the Lab Admin users.

Assign security roles

Navigate to the Power Platform Admin Center https://aka.ms/ppac

Select the **Device Ordering Development** environment.

Click on **Settings**.

Expand **Users + permissions**.

Select **Security Roles**.

Select the **System Customizer** role.

Click **+ Add people**.

Enter **labadmin**

Select a Lab Admin user.

Repeat and select all Lab Admin users.

Click on **Add**.

You are now ready to run the ALM lab.

demo: title: 'Demo: Deploying an ARM Template' module: 'Module 1: Exploring Azure Resource Manager'

Demo: Deploying an ARM Template

Instructions

1. Quisque dictum convallis metus, vitae vestibulum turpis dapibus non.

    1. Suspendisse commodo tempor convallis.

    2. Nunc eget quam facilisis, imperdiet felis ut, blandit nibh.

    3. Phasellus pulvinar ornare sem, ut imperdiet justo volutpat et.

2. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos.

3. Vestibulum hendrerit orci urna, non aliquet eros eleifend vitae.

4. Curabitur nibh dui, vestibulum cursus neque commodo, aliquet accumsan risus.

    ```
    Sed at malesuada orci, eu volutpat ex
    ```

5. In ac odio vulputate, faucibus lorem at, sagittis felis.

6. Fusce tincidunt sapien nec dolor congue facilisis lacinia quis urna.

    Note: Ut feugiat est id ultrices gravida.

7. Phasellus urna lacus, luctus at suscipit vitae, maximus ac nisl.

    - Morbi in tortor finibus, tempus dolor a, cursus lorem.
    - Maecenas id risus pharetra, viverra elit quis, lacinia odio.
    - Etiam rutrum pretium enim.

8. Curabitur in pretium urna, nec ullamcorper diam. **Module 8: Application Lifecycle Management**
   ==========

Lab Scenario

In this hands-on lab, you are an administrator 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 Apps 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.

Exercise 1: Initialize Azure DevOps

In this exercise, you will be signing up for an Azure DevOps account and configuring the Power Apps build tools for the account.

Note: If you already have Azure DevOps outside of this course and this environment, you **CANNOT** use that here. You will need to follow the below instructions to sign up.

Task 1: Signup for Azure DevOps

1. Log on to https://dev.azure.com and click **Sign into Azure DevOps;** use the credentials from your instructor.

2. Provide your credentials and click **Next**.

3. Provide a **Password** and click **Sign in**.

4. Click **Continue**.

5. Provide a unique **Azure DevOps Organization** name such as lastnameMMYY, select a location closest to your tenant, enter captcha and click **Continue**. *Replace lastname with your last name, MM current month and YY current year.*

   Note: For some users, the DevOps Organization is automatically created using your username. This is OK and you do not need to rename the Organization.

   Note: For some users, this page has a heading ���We need a few more details���. Ensure that you enter the organization name and not miss this step.

6. 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.* Select **Private**for **Project Visibility**. This will take 3-7 seconds to configure and navigate to the Project���s Home page.

7. 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.

8. Select **Repos | Files**.

9. 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.

10. Scroll down and check the **Add a Readme** checkbox and then click **Initialize**.

Task 2: Configure Power Apps Build Tasks

1. Log on to https://marketplace.visualstudio.com/azuredevops

2. Search for **PowerApps**.

3. Select **PowerApps BuildTools**.

4. Click **Get it Free**.

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

6. Click **Proceed to Organization**. This will open a new tab for the project.

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

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

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

10. Navigate to the **Permissions** tab then select **Project Collection Build Service Accounts**

11. Locate **Contribute** and set it to **Allow**.

12. Search for **Project Collection Build Service** and select the Project Collection without Accounts at the end of the name.

13. Locate **Contribute** and set it to **Allow**.

Exercise 2: Build 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.

Task 1: Export Solution

1. Create a Build Pipeline.
    1. Click to expand **Pipelines**.
    2. Click **Create Pipeline**.
    3. Click **Use the Classic Editor to create a pipeline without YAML.**
    4. Click **Continue**. Do not change the default values.
    5. Select **Empty Job**.
    6. Click **Save and Queue** and select **Save**.
    7. Click **Save**.
2. Add PowerApps Tool Installer task. This is found under **Deprecated Tasks Note:** The PowerApps Tool Installer needs to be run before any other PowerApps build tasks.
    1. Click **+ Add Task** to **Agent Job 1**.
    2. Search for **PowerApps Tool** hover over select **PowerApps Tool Installer** and click **Add**.
3. Add PowerApps Export Solution task. This is found under **Deprecated Tasks**
    1. Search for **Export**

2. Hover over **PowerApps Export Solution** and click **Add**.

4. Open PowerApps Export Solution.

    1. Select the **PowerApps Export Solution** task.

5. Get your development environment URL

    1. Start a new browser window or tab and log on to https://admin.powerplatform.microsoft.com. Use the same credentials that you are using in Azure DevOps to log in.

    2. Select **Environments** and click to open the **Device Ordering Development** environment.

    3. Copy the **Environment URL** and keep it in your clipboard. Keep this URL on a notepad.

    Note: If the **Device Ordering Development** environment is not listed in the Power Platform Admin Centre, log on to https://make.powerapps.com/ and click on **Apps**. Click on **Device Procurement** app to play the app. Copy the **URL** from the browser app up to and including dynamics.com.

    Note: To make sure that the required solution exists, log on to https://make.powerapps.com/ and make sure you are in the **Device Ordering Development** environment. Select **Solutions** from the left menu and you should see the **Contoso Device Order Management** unmanaged solution.

    1. Close the **Power Platform Admin** browser window or tab.

    2. Close the **Maker Portal** browser window or tab.

6. Create a **Generic Service Connection.** Service Connections are how the build tasks know what environment URL and user credentials to use to access the Common Data Service environments.

    1. Go back to the **Pipeline**.

    2. Make sure you still have the **PowerApps Export Solution** task selected.

3. Click **Manage** next to Service Connection. This will open a new window.

4. Click **Create Service Connection**.

5. Search and Select **Generic** and click **Next**.

6. Paste the **Environment URL** you copied (the URL should begin with https://), provide your credentials, provide the **Connection Name** as **Dev Connection**, and click **Save**.

7. Make sure that the connection is created and then close the **Service Connections** browser window or tab.

7. Select the Generic Service Connection you created as the Power Apps Environment URL.

    1. Go back to the **Build Pipeline** tasks and make sure you still have PowerApps Export Solution task selected.

    2. Locate the **PowerApps Environment URL** field and click **Refresh**.

    3. Select the **Generic Service Connection** you just created named **Dev Connection**.

    4. Enter **\$(SolutionName)** for **Solution Name**, **\$(Build.ArtifactStagingDirectory)\\$(SolutionName).zip** for **Solution Output File**. Ensure that there are no white spaces while entering the above values.

    5. Click **+ Add Task** again.

    6. Add another **Export Solution** task.

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

    8. Select the connection you created, enter **\$(SolutionName)** for **Solution Name**, **\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip** for **Solution Output File**. Ensure that there are no white spaces while entering the above values.

9. Check the **Export as Managed Solution**.

8. Add an Unpack task. This task will take the solution zip file and expand it into a file for each solution component. This is found under **Deprecated Tasks**.

    1. Click **+ Add Task**.

    2. Search for **Unpack**.

    3. Hover over **PowerApps Unpack Solution** and click **Add**.

9. Provide Unpack settings information.

    1. Select the **Unpack** task.

    2. Enter **\$(Build.ArtifactStagingDirectory)\\$(SolutionName).zip** for **Solution Input File**, **\$(Build.SourcesDirectory)\\$(SolutionName)** for **Target Folder**. Ensure that there are no white spaces while entering the above values.

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

10. Allow scripts to access the OAuth Token. This will allow the commands you will add 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. Select **Agent Job 1**.

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

    3. Click **Save and Queue** and select **Save**.

    4. Click **Save** again.

11. Add Command Line task.

    1. Click **+ Add a Task**.

2. Search for **Command Line**.

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

12. Add Scripts to the Command Line task. This task will be used to check in the solution file changes to the repo.

    1. Select the Command Line task.

    2. Paste the script below in the **Script** text area. Replace **user\@myorg.onmicrosoft.com** with your user credentials.

```
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
```

13. Add Solution Name variable.

14. Select the Variables tab.

15. Click **+ Add.**

16. Enter **SolutionName** for **Name** and **ContosoDeviceOrderManagement** for **Value**. Ensure that there are no white spaces while entering the above values.

17. Click **Save and Queue** and select **Save**.

18. Click **Save** again.

19. Allow Contribute for Build Service.

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

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

22. Navigate to the **Permissions** tab

23. Select **Device Management lastnameMMYY Build Service** user.

Note: If the Azure DevOps Organization was generated automatically, the name maybe slightly different.

24. Locate **Contribute** and set it to **Allow**.

Exercise 3: Test the Pipeline

In this exercise, you will test the build pipeline you created.

Task 1: Run the Pipeline

1. Open the build pipeline.
    1. Log on to https://dev.azure.com/ and click to open the **Device Management lastnameMMYY** project.
    2. Select **Pipelines**.
    3. Click to open the pipeline.
    4. Click **Run Pipeline**.
    5. Click **Run** again and wait.
    6. Wait for the job to complete. The job should run and succeed. Click to open it.
2. Review the Repository.
    1. Select Repos.
    2. You should see **ContosoDeviceOrderManagement** folder. Click to open the folder.
    3. You may examine the content of each folder.

Exercise 4: Build Manage 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 Common Data Service environment to check for missing dependencies. You could also add build tasks to run tests against your solution as well as run PowerApps 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. Select **Pipelines**.
2. Click **New Pipeline**.
3. Click **Use Classic Editor**.
4. Click **Continue**.
5. Click **Empty Job.**
6. Enter **Build Managed Solution** for **Name**, click **Save and Queue** and select **Save**.
7. Click **Save** again.
8. Click **Add a Task to Agent Job 1**.
9. Search for **PowerApps Tool**, hover over **PowerApps Tool Installer** and click **Add**. This is found under **Deprecated Tasks**.

10. Search for **PowerApps Pack**, hover over **PowerApps Pack Solution** and click **Add**. This is found under **Deprecated Tasks**.

11. Select the **PowerApps Pack Solution** task.

12. Enter **\$(Build.SourcesDirectory)\\$(SolutionName)** for **Source Folder of Solution to Pack**, enter **\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip** for **Solution Output Folder**, and select **Managed** for **Type of Solution**. Ensure that there are no white spaces while entering the above values.

13. Select the **Variables** tab and click **+ Add**.

14. Enter **SolutionName** for **Name** and **ContosoDeviceOrderManagement** for **Value**. Ensure that there are no white spaces while entering the above values.

15. Select the **Tasks** tab and click **Add a Task**.

16. Search **Publish Pipeline**, hover over **Publish Pipeline Artifacts** and click **Add**. Publishing the solution as an artifact will make it available for the release pipeline you will build.

17. Select the **Publish Pipeline Artifact** task.

18. Enter **\$(Build.ArtifactStagingDirectory)\\$(SolutionName)_managed.zip** for **File or directory Path** and enter **drop** for **Artifact Name**. Ensure that there are no white spaces while entering the above values.

19. Click **Save and Queue** and select **Save and Queue**.

20. Click **Save and Run**.

21. Wait for the run to complete.

22. Job tasks should run and succeed. Click to open the job.

23. Click on the **Artifact** link.
    Note: The artifcat link can be found by selecting **Agent job 1**. If you do not see the artifacts, refresh the browser and it will appear.

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

Exercise 5: Build a 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 test environment.

Task 1: Create Connection and Release to Prod

1. Log on to https://dev.azure.com/
2. Click to open the **Device Management** project.
3. Click **Project Settings.**
4. Go to the **Pipelines** area and select **Service Connections**.
5. Click **New Service Connection**.
6. Select **Generic** and click **Next**.
7. In a new tab, log on to https://admin.powerplatform.microsoft.com
8. Select **Environments** and click to open the **Prod - labadminx** environment where x is a number.
9. Right click on the **URL** and select **Copy Link**.
10. Close the **Admin Center** browser tab.
11. Paste the **URL** you copied to the **Server URL**, provide your credentials, **Test Connection** for **Server Connection Name**, and click **Save**.
12. Select **Pipelines** -> **Releases**.

13. Click **New Pipeline**.

14. Select **Empty Job**.

15. Enter **Test** for **Stage Name** and click **+ Add** under **Artifacts**.

16. Select **Build Managed Solution** for **Source** and click **Add**.

17. Select the **Tasks** tab and click **+ Add Task**.

18. Search for **PowerApps Tool**, hover over **PowerApps Tool Installer** and click **Add**. This is found under **Deprecated Tasks**.

19. Search for **PowerApps Import**, hover over **PowerApps Import Solution** and click **Add**. This is found under **Deprecated Tasks**.

20. Select the **PowerApps Import Solution** task.

21. Select the **Test Connection** for Environment and enter **\$(System.DefaultWorkingDirectory)/_Build Managed Solution/drop/\$(SolutionName)_managed.zip** for **Solution Input File**. **Note**: There are spaces around the word Managed.

22. Select the **Variables** tab and click **+ Add**.

23. Enter **SolutionName** for **Name**, **ContosoDeviceOrderManagement** for **Value**, and click **Save**. Ensure that there are no white spaces while entering the above values.

24. Click **OK**.

25. Click **Create Release**.

26. Click **Create**.

27. Select **Releases** and click to open the release.

28. Wait for the release tasks to complete. The release pipeline should succeed.

29. Log on to https://make.powerapps.com and select the **Prod - labadminx** environment where x is a number.

30. Select **Solutions**. You should see the managed solution you published from the release pipeline.

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

32. You should be able to see the **You cannot edit managed** solution message.

33. Select Apps. You should see **Device Ordering App** Canvas application and **Device Procurement** Model-Driven-Application.

Table of Contents