



Module 6

Power Platform Application Lifecycle Management

Microsoft Customer Success Unit



CONDITIONS AND TERMS OF USE:

© Microsoft Corporation. All rights reserved.

You may use these training materials solely for your personal internal reference and non-commercial purposes. You may not distribute, transmit, resell or otherwise make these training materials available to any other person or party without express permission from Microsoft Corporation. URL's or other internet website references in the training materials may change without notice. Unless otherwise noted, any companies, organizations, domain names, e-mail addresses, people, places and events depicted in the training materials are for illustration only and are fictitious. No real association is intended or inferred. THESE TRAINING MATERIALS ARE PROVIDED "AS IS"; MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED IN THESE TRAINING MATERIALS.

Introduction

In this lesson, you will learn the following:

- Understand Power Platform ALM
- Power Platform Deployment Lifecycle
- Deploy Resources using a Package or a Solution

Application Lifecycle Management

- The Power Platform administrator role should consider managing environments' resources including apps, flows, connections and data sources.
- Building an ALM plan for app makers is very critical to an enterprise to help manage, maintain and monitor these resources in each environment.
- Let's look first at things to consider as an administrator to help guide the application through its lifecycles from new to production and then at ongoing maintenance and enhancements.
- For purposes of this section, application refers to the whole set of components from Power Apps canvas or model-driven apps, flows and any Common Data Service customizations.

Business Apps and Automation Requests Central Hub

- For large organizations users' requests for new business apps and automation should be managed from a centralized location where operations and solutions architects should advise whether this solution should fulfill missing business needs, or an organization already has this feature available.
- This request should run through a predefined process to collect certain approvals and accepting request terms and conditions defines future responsibilities.
- Power Platform features can be used to build such a central hub.
- Upon requests approval, a concerned user or team can start implementing the solution at the organization or departmental development environment.
- Consider all points mentioned in a previous slide regarding New Applications.

Managing Power Platform Environments



Power Platform administrators should define ALM for their resources based on organization/department level that is based on actual business needs and available licenses.



It is highly recommended to keep each resource hosted based on their categories as follows:

Organization Resources
the entirety of the tenant users.

Departmental Resources
business department users.



This type of classification is helping administrators to define each environment severity and keep maintenance plans simplified.

Validate and Test Resources



Resource makers and business owners should validate implemented resources before final deployment to production and before sharing these resources with business users.



Testing processes could be different based on resource scope and complexity.



User acceptance testing (UAT) is the common contract between resource developers and business owners

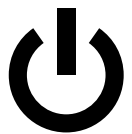
Managing Deployed Resources Support



It is highly recommended to consider the Contact Support feature as an approval criteria for production deployment.



For Power Apps resources, an action button or screen should be created to contact app the support team using a button or sending an email link.

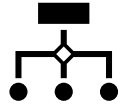


For Power Automate resources, a report can be created using Power Apps or Power BI to show flow runs and contact owners directly.

Determine Deployment Method



Power Platform provides multiple methods for deploying resources.



The Power Platform administrator should determine which deployment method is more suitable to select or use a method for all deployment scenarios.



Power Platform deployment methods:

- Using Packages
- Using Solutions

Which Resources Can Be Migrated Using a Package

When you export an app, the dependent resources of the app will also be included into the package.

Initially only a subset of all possible resource types will be supported as outlined in the table below.

Resource type	Supported	Import options
App	<u>Yes, for canvas apps</u>	There are two options to import an app into an environment: Create new and Update
Power Automate	<u>Yes</u>	There are two options to import a flow into an environment: Create new and Update Note: All resources that the flow depends on will also be included within the app package that is exported and will need to be configured when the package is imported.
Custom Connectors	No	Not supported at the moment. If you have an app that relies on a custom connector, your only current option is to manually re-create or update the connector in your target environment
Connections	No	<u>We do not currently</u> support exporting the connection or credentials as a part of the package.
Common Data Service Customizations	No	Exporting Common Data Service customizations is no longer supported as a part of packaging. <u>This is now supported</u> through export and importing the environment solutions.
Gateways	No	Gateways are only supported in the Default (and Preview) environments, so export/migration is not supported.

Deployment Using Solutions

- In Power Platform, solutions are leveraged to transport apps and components from one environment to another or to apply a set of customizations to existing apps.
- A solution can contain one or more apps and flows as well as other components such as site maps, entities, processes, web resources, option sets, and more.
- Some components are nested within other components, for example, an entity contains forms, views, charts, fields, entity relationships, messages, and business rules.
- Each of those components requires an entity to exist.
- A field can't exist outside of an entity.
- Solutions need environments with Dataverse (CDS) installed to be used – creating environment with Dataverse (CDS) will explained later in this module.

Unmanaged Solutions

- An unmanaged solution is one that is still under development or isn't intended to be distributed. When the unmanaged solution is complete and you want to distribute it, export it and package it as a managed solution.
- Solutions created by default as unmanaged are then exported as managed or unmanaged.
- While a solution is unmanaged, you can continue to add and remove components to and from it.
- You can export an unmanaged solution to transport unmanaged customizations from one environment to another.
- When an unmanaged solution is deleted, only the solution container of any customizations included in it is deleted. All the unmanaged customizations remain in effect and belong to the default solution.
- You can't delete the components by uninstalling the solution.
- When you import an unmanaged solution that contains components that you have already customized, your customizations will be overwritten by the customizations in the imported unmanaged solution and can't be undone.

Managed Solutions

- A managed solution is a complete solution that is intended to be distributed and installed.
- To create a managed solution, you choose the As managed option when you export the solution.
- You cannot edit the components of a managed solution
- You cannot export a managed solution.
- Can be uninstalled after it is imported which will be deleted from the environment
- You cannot import a managed solution into the same environment that contains the original unmanaged solution. To test a managed solution, you need a separate environment to import it into.

Lab: Module 6

Exercise 1 - Deploy Resources Between Environments Using a Package

Exercise 2 - Deploy Resources Between Environments Using a Solution



