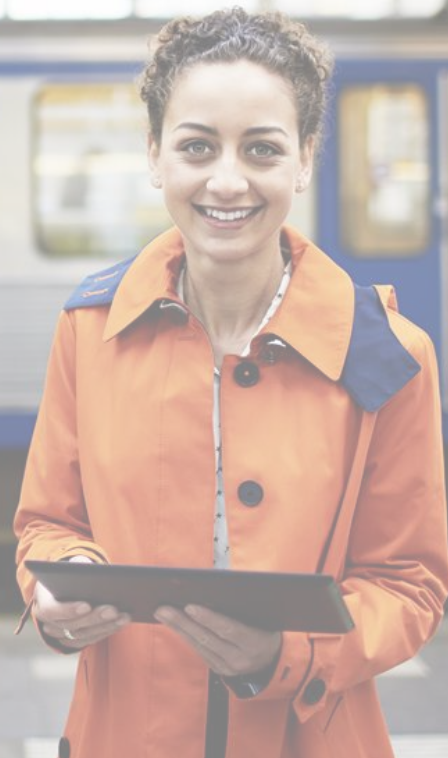


FastTrack for Dynamics 365 CE Tech Talk Series

Application Lifecycle Management



Microsoft



Agenda/Objectives

1 Overview

2 Environment/Storage Planning/Management

3 Solution Management

4 Deploying Solutions

5 Tools for Automation

6 Automation Demo

Solution Development is addressed in our Solution Development Tech Talk. If you're not familiar with our Solution Development process, you should watch that before watching this presentation, so you have better context for some of these topics.

Duration

- 1 hour

Outcomes

- Basic concepts of Application Lifecycle Management with Dynamics 365 Model-driven Apps
- Developing an ALM Strategy approach
- Understand how having the right number and type of environments can simplify development
- Solution basics, the different types and how to manage, package, validate and deploy them
- Include new Power Platform components in your solutions
- Basic automation of your DevOps model using Azure Build Tools
- Testing Automation with EasyRepro



Overview



Overview – Application Lifecycle Management



Application Lifecycle Management or **ALM** represents the governance, development, deployment and on-going maintenance of computer applications.

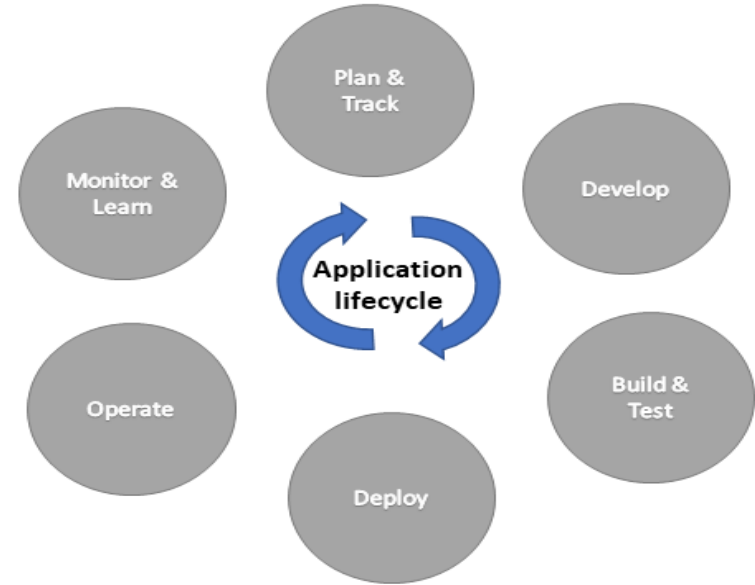
- ✓ We will consider *computer applications* to represent the Dynamics 365 Model-driven Apps and Solution(s) you create to enable and enact your business processes and outcomes and some of the **processes, tools** and **best practices** available to assist you.



Overview – Developing an ALM Strategy



- Consider your development approach
- Types of development required (web/.Net/cloud)
- Development team skills/size/structure
- Number/type of and storage for CDS instances
- Source Control/Backlog Management



- Testing Model
- Deployment Model
- Operate/Maintain Model
- Automation tooling and approach
- Governance Model



Overview – Processes, Tools and Best Practices

Processes

ALM Strategy

Governance

Environment Management

Solution Management

Source Control

Deployment

Continuous Integration (with Tools)

Tools

Power Platform

Power Apps

Solution Checker

Configuration Migration Tool

Git / Visual Studio

Solution Packager

Package Deployer

Azure DevOps Build Tools

EasyRepro Test Automation

Best Practices

Strategy and Governance

Solution Management

Deploying Solutions

Build Automation

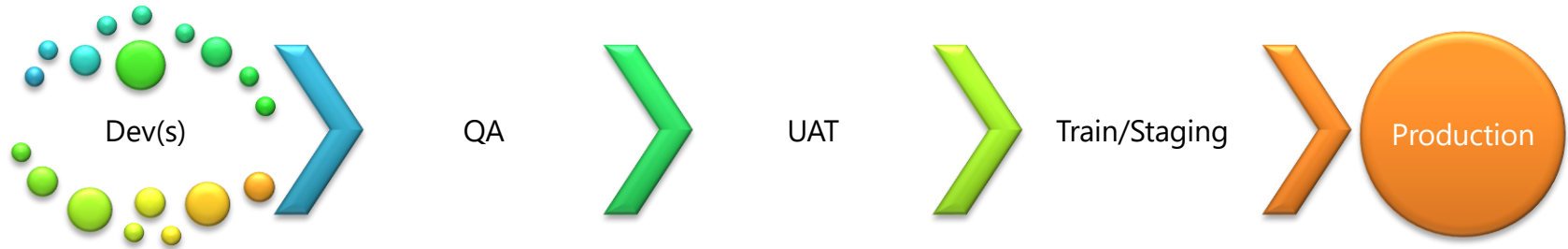
Source Control



Environment/Storage Planning & Management



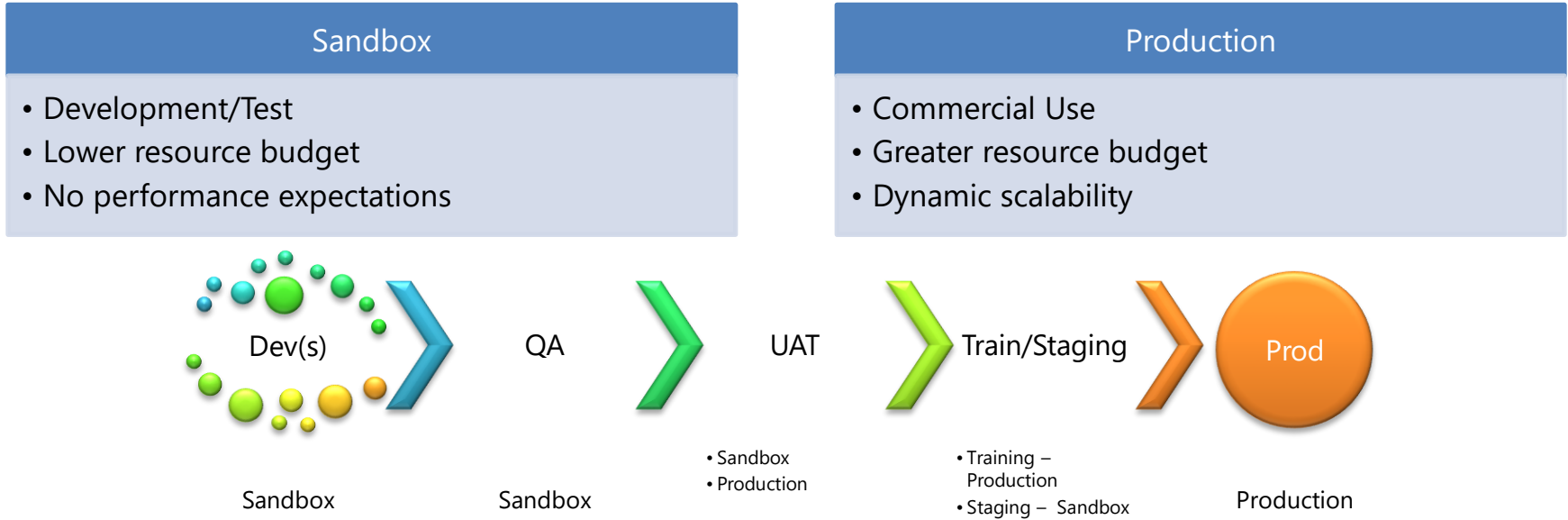
Environment Planning – Instance Strategy



- ✓ Define the instance requirements necessary to support the ideation, build and operation your business
- ✓ Should facilitate project decisions such as the development model, go-live approach, where to conduct a POC (Proof of Concept), UAT (User Acceptance Testing), Hot Fix/Break Fix, etc., identifying what instances are needed and when.
- ✓ Dynamics 365 Model-drive App-enabled instances provide basic capability for you to migrate solutions between instances
- ✓ Consider using a **Just-in-Time (JIT) instance deployment model** and build from source with the correct data sets to accomplish the goal of the instance required
- ✓ Using a JIT model facilitates **source control centrality**, eliminates dependences on specific instances and allows you to validate your deployment process each time you deploy



Environment Planning – Instance Types

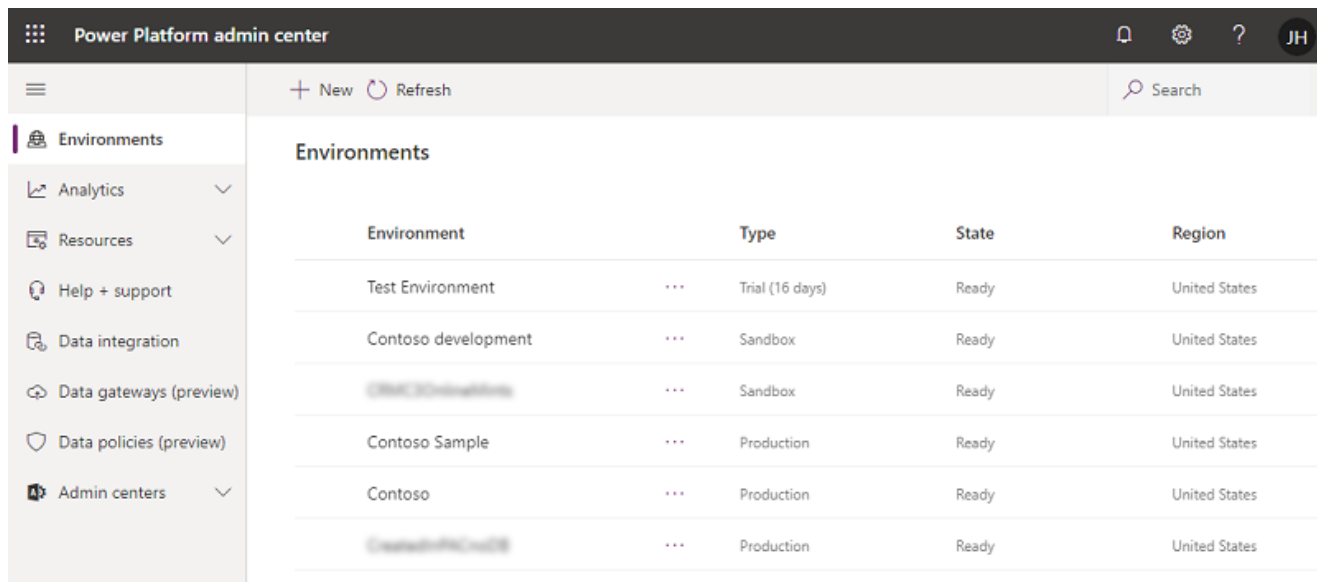


- There are other environment types but only these are relevant for Dynamics 365 Model-driven Apps



Environment Planning – Instance Management

- Instances are CDS environments created to be used with Dynamics 365 Model-driven Apps
- Can be created, configured and managed from the Power Platform Admin Portal.



The screenshot shows the Power Platform Admin Center interface. The left sidebar contains navigation links: Environments (selected), Analytics, Resources, Help + support, Data integration, Data gateways (preview), Data policies (preview), and Admin centers. The main content area is titled 'Environments' and features a table with columns: Environment, Type, State, and Region. The table lists several environments, including 'Test Environment', 'Contoso development', 'Contoso Sample', and 'Contoso'. Each row includes a three-dot menu icon for actions.

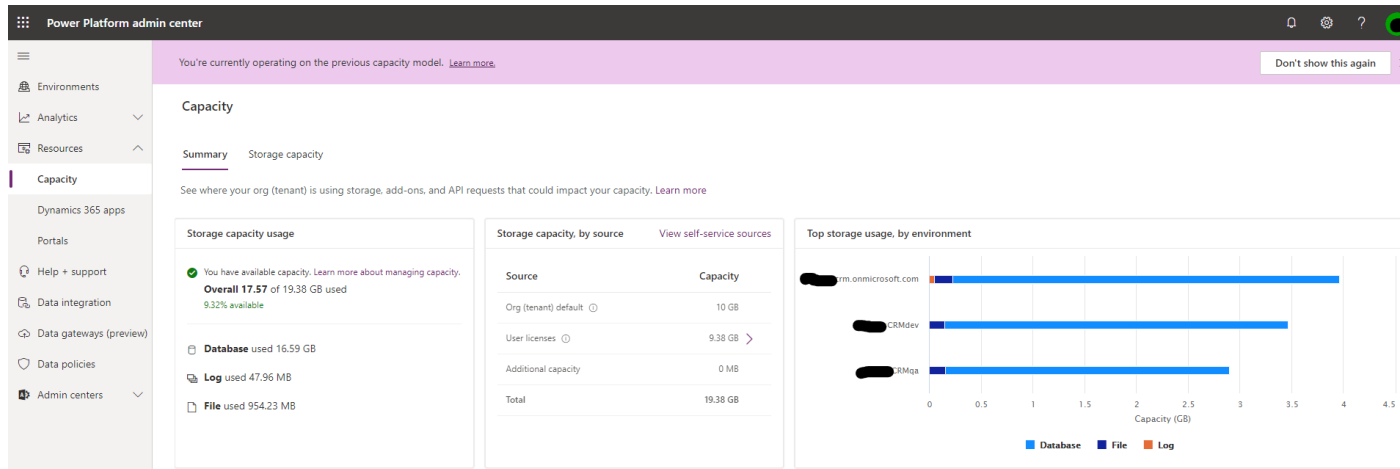
Environment	Type	State	Region
Test Environment	Trial (16 days)	Ready	United States
Contoso development	Sandbox	Ready	United States
Contoso Sample	Production	Ready	United States
Contoso	Production	Ready	United States

For more information on Tenant and Instance Administration, you can watch our *Tenant Administration Tech Talk* and you can learn more here <https://docs.microsoft.com/en-us/power-platform/admin/admin-documentation>



Storage Planning – Determining Needs

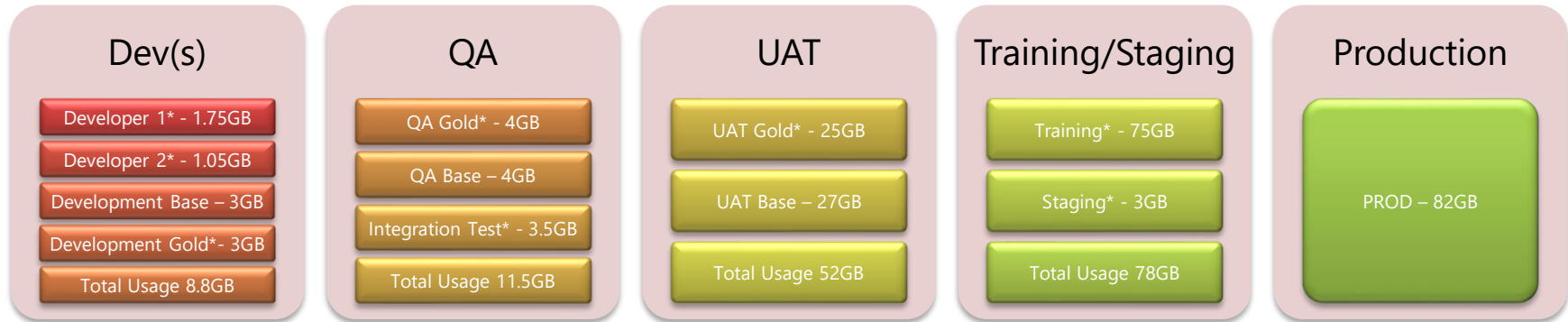
- Determining the necessary storage capacity is another key element of a successful ALM strategy.
- Understanding how much storage your ALM process requires will help you as you plan for an initial deployment as well as growth over time.
- Make sure you understand your storage model (Legacy/New)
- You can see your model, overall storage capacity, total usage and usage by instance at <https://admin.powerplatform.microsoft.com/resources/capacity>



To understand more about storage management and how to properly administrate it, see <https://docs.microsoft.com/en-us/power-platform/admin/whats-new-storage>



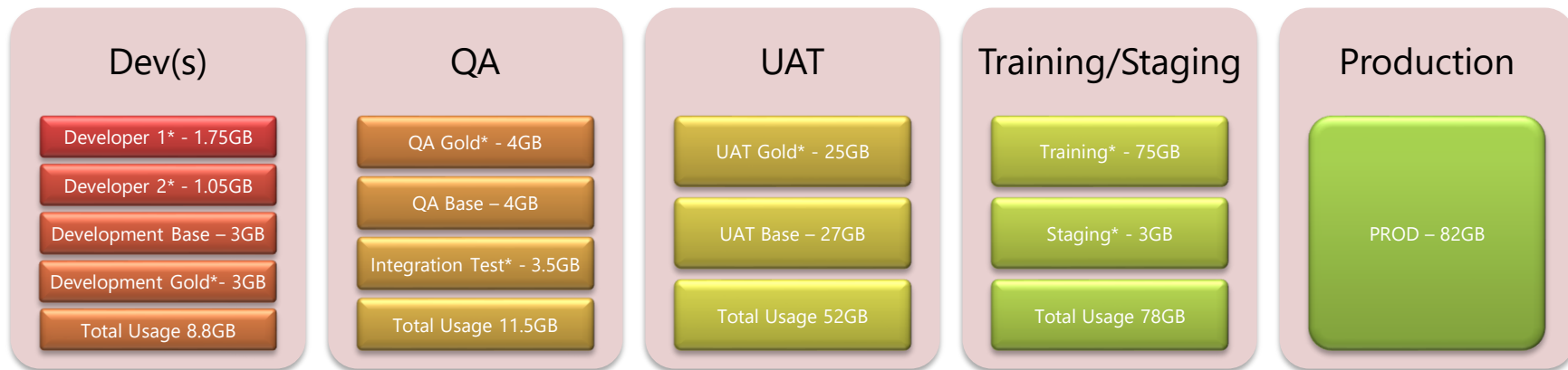
Storage Planning – Example



- Base environments are the current *active* environments. Optional environments (marked with *) can be employed to either expand capability or maintain PROD-equivalent copies
- Dev environments should be kept relatively small (<1.5Gb of total data in each)
- QA environments should typically have only enough data to validate current scenarios. Additional data to validate new scenarios should be added as those scenarios are created and migrated to the environment
- UAT (User Acceptance Testing) environments should only maintain sufficient data to allow users to test end-to-end scenarios inclusive of any new functionality as part of the release being tested/validated.
- Training/Staging environments have completely different requirements from each other but both live in the same tier.
 - Training environments may contain a significant of data to ensure all user scenarios are adequately represented.
 - Staging environments should not require any application data and are typically used as a *production-equivalent* place holder that can be leveraged as a safe placeholder of the production solution.



Storage Planning – Example Lessons/Strategy



- Total Tenant Storage consumed is 232.3GB
- Optional instances consume **116.3GB!**
- Clearly understand and define your approximate storage needs BEFORE you start deploying a complex instance model. Addressed in more detail in our Tenant Administration Tech Talk
- Ensure your storage management strategy includes accounting for growth over time and enough additional capacity to address potentially unforeseen needs (POC's, significant unplanned data growth, etc.)
- Purchase sufficient additional capacity your model dictates well in advance of needing it to protect yourself from delays in proceeding with your implementation



Solution Management



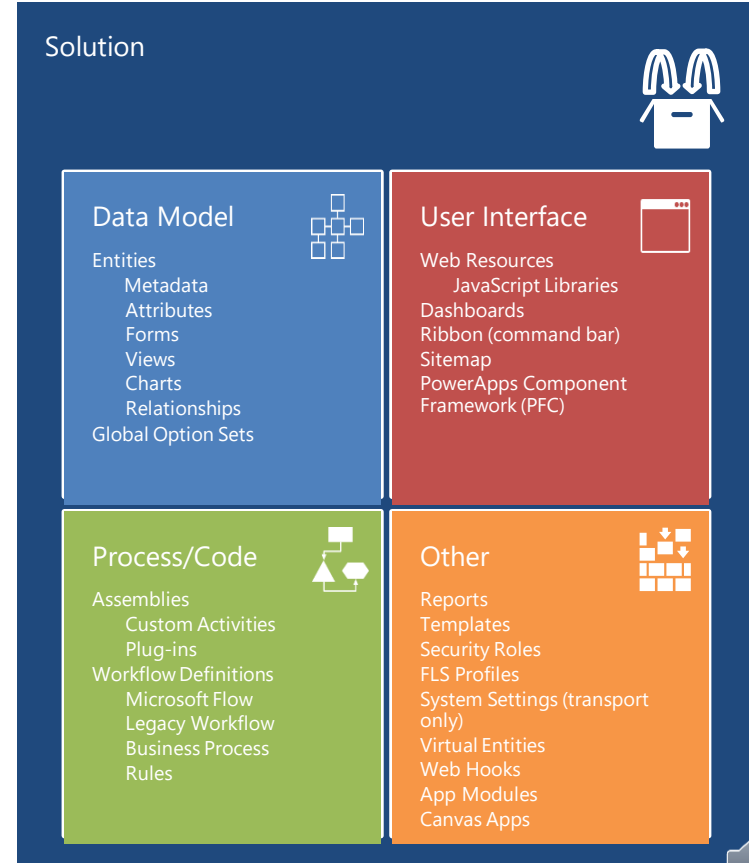
The Basics of a Solution Refresher/Background



Solution Basics - What is a Solution?

- ✓ Solutions are how developers author, package, and maintain units of software that extend Dynamics 365 Customer Engagement
- ✓ Solutions must be associated with a Publisher
- ✓ Solution Files are zipped XML configuration files containing all definitions, configurations, customizations and any other necessary components related to that solution.

For more information, please see <https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/introduction-solutions>





Solution Basics – Solution Types

Unmanaged	Managed
<p>A logical container of customizations existing on top of the system and managed solutions</p> <p>Unmanaged solutions are developed</p> <ul style="list-style-type: none">• They allow direct customization to components• They hold references to customization and components but do not own them• They do not delete or remove components if the solution is deleted• The <i>Default</i> solution is the highest unmanaged solution layer.• NOTE: We do not recommend you develop in or deploy the <i>Default</i> solution	<p>A “version complete” unmanaged solution that has been exported as managed</p> <p>Managed solutions are deployed</p> <ul style="list-style-type: none">• The identity of the solution cannot be modified• Components cannot be added or removed• Exist within their own discrete layer• Combined with other layers and the system layers to present application behavior to the user• Can be uninstalled (deleted) and does delete components and data they directly own as well as related data they don’t directly own• Upgraded/Updated within its own discrete layer allowing ISVs or multiple departments to provide multiple applications to a shared environment



Solution Basics - Publishers

→ Publisher Main View ▾

 New |  | More Actions ▾

<input type="checkbox"/>	Display Name	Name	Prefix	Description	Phone	Email	Website
	CDS Default Publisher	Crb2b7d	crf24				
	Default Publisher for [redacted]	DefaultPublisher [redacted]	new	Default publisher for this organization			
	Microsoft First Party	microsoftfirstparty		Publisher for Microsoft First Party Solutions			
	Microsoft Forms Pro	microsoftformspro	msfp		1-800-Microsoft(642-7676)		
	Dynamics 365	dynamics365customerengage...	msdynce	Dynamics 365	1-800-Microsoft (642-7676)		http://crm.dynamics.c...
	Microsoft Dynamics 365	microsoftdynamics	msdyn	Provides access to Dynamics 365 App for Outloo...	1-800-Microsoft(642-7676)		http://crm.dynamics.c...

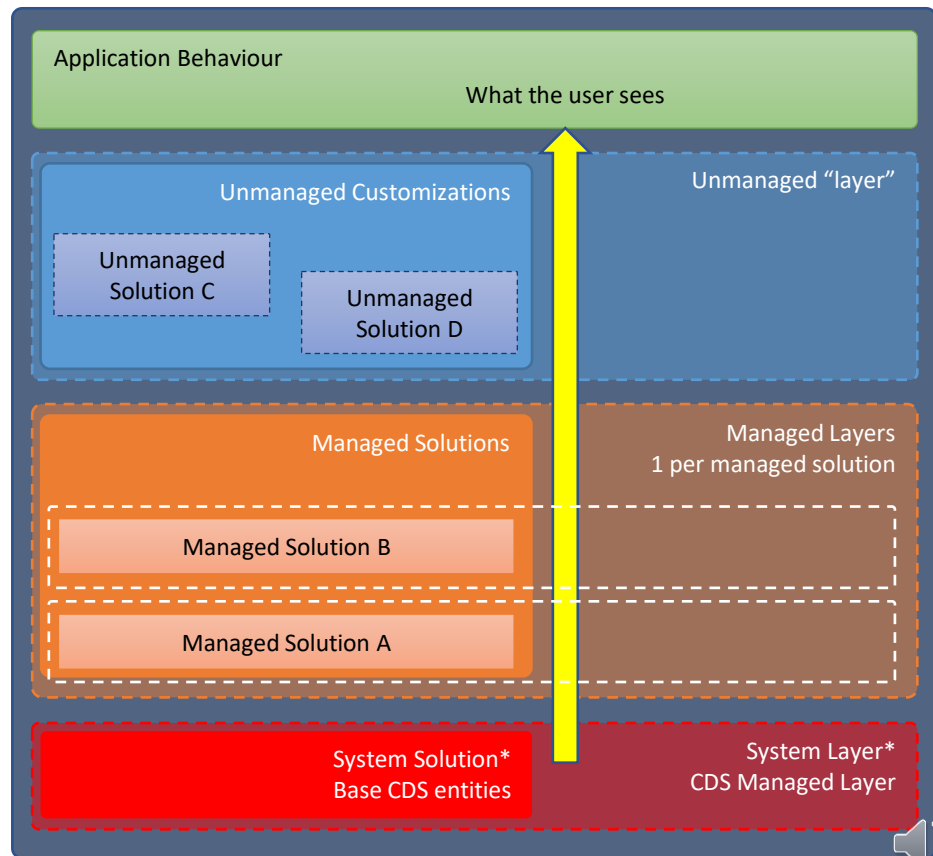
- A Solution-related entity that represents the “owner” of a published solution and the assets contained within.
- You can see your publishers in your instance within UCI by selecting the *Settings Wheel – Advanced Settings*, then under the *Settings* categories, choose *Customization* and then *Publishers*
- Several publishers are created by default when you get a new instance of Dynamics 365 (see above). **DO NOT use or change any of the default publishers provided with your instance.**
- You can create your own publishers using the **New** button. It does not have to be associated to a system user but only System Customizers and System Administrators can use them.
- Publishers may also be added for 3rd Party (ISV or Partner) Managed solutions you deploy.
- This is a critical tool in helping to manage governance of your Solution Development



Solution Basics - Layering

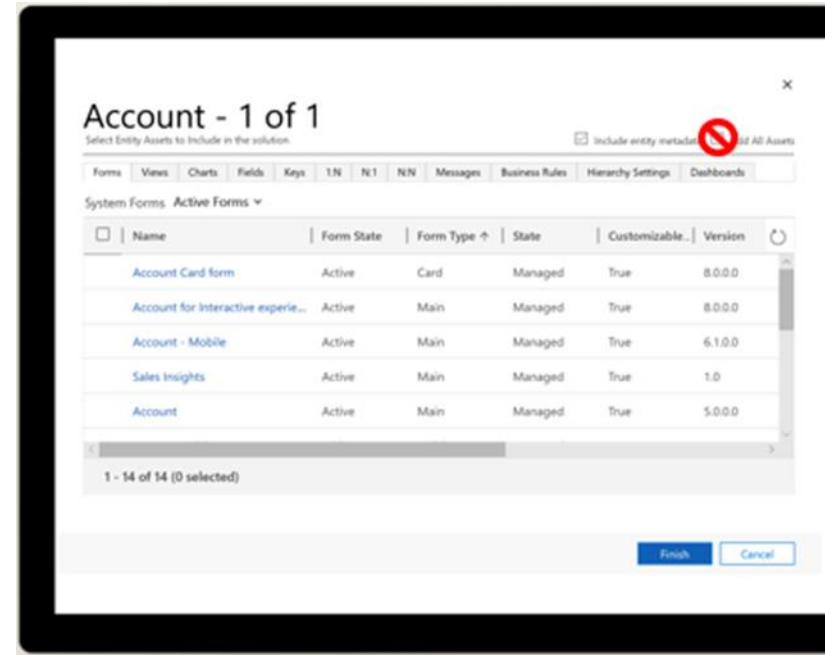
- ✓ Layers describe the dependency chain of a component from the root solution introducing it, through each solution that extends or changes the component's behavior
- ✓ Layers are created through extension of an existing component (taking a dependency on it) or through creation of a new component or version of a solution.
- ✓ Managed Solutions will always update the previous version of itself and will live at that same solution layer
- ✓ Unmanaged Solutions will always sit above the Managed Solution layer(s)
- ✓ Configurations not bound to a solution live at the topmost Unmanaged layer although there is no actual layering with unmanaged configurations/solutions
- ✓ **Important:** Layers should always be considered on a per-component basis. Although the diagram to the right is typically drawn to convey the effect of a solution on another solution, technically this is always at a component or sub-component level.

For more information please see <https://docs.microsoft.com/en-us/power-platform/alm/solution-layers-alm>

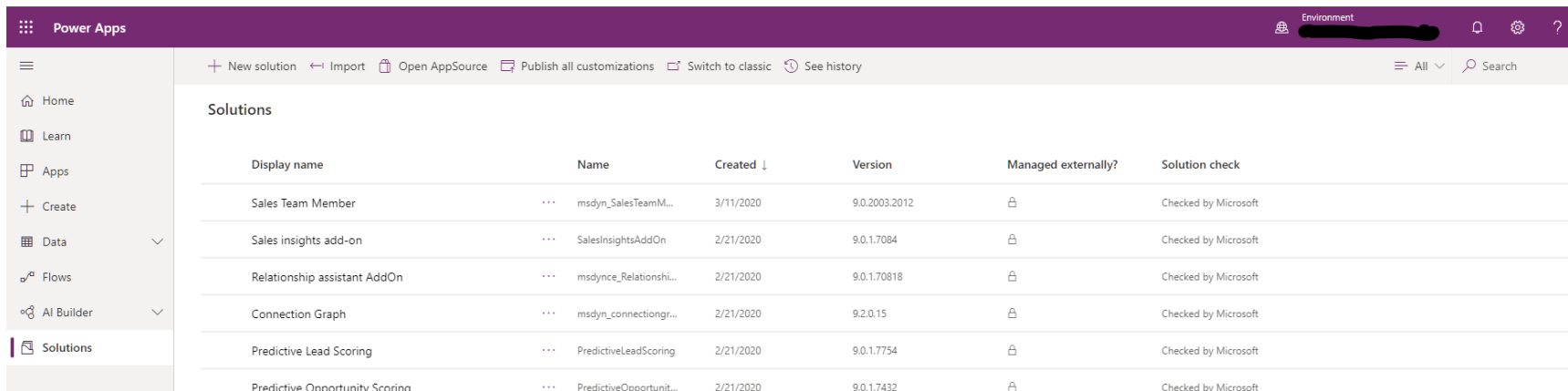


Solution Basics - Segmentation

- ✓ Segmentation is the means by which you can physically partition solution components to reduce both the complexity and size of a given solution
- ✓ Segmentation will be established when you select/add only the specific components you need while creating or updating a solution
- ✓ **Avoid including all assets for any entity that already exists in the target environment of your solution**



Solution Basics – Manage with Power Apps



The screenshot shows the Power Apps 'Solutions' page. The top navigation bar is purple with the 'Power Apps' logo on the left and an 'Environment' dropdown on the right. Below the navigation bar is a command bar with options: '+ New solution', '← Import', '📁 Open AppSource', '📄 Publish all customizations', '📂 Switch to classic', and '🕒 See history'. On the right of the command bar are 'All' and 'Search' filters. The left sidebar contains a menu with 'Home', 'Learn', 'Apps', 'Create', 'Data', 'Flows', 'AI Builder', and 'Solutions' (which is highlighted). The main area is titled 'Solutions' and contains a table with the following columns: 'Display name', 'Name', 'Created', 'Version', 'Managed externally?', and 'Solution check'.

Display name	Name	Created	Version	Managed externally?	Solution check
Sales Team Member	msdyn_SalesTeamM...	3/11/2020	9.0.2003.2012	🔒	Checked by Microsoft
Sales insights add-on	SalesInsightsAddOn	2/21/2020	9.0.1.7084	🔒	Checked by Microsoft
Relationship assistant AddOn	msdynce_Relationshi...	2/21/2020	9.0.1.70818	🔒	Checked by Microsoft
Connection Graph	msdyn_connectiongr...	2/21/2020	9.2.0.15	🔒	Checked by Microsoft
Predictive Lead Scoring	PredictiveLeadScoring	2/21/2020	9.0.1.7754	🔒	Checked by Microsoft
Predictive Opportunity Scoring	PredictiveOpportunit...	2/21/2020	9.0.1.7432	🔒	Checked by Microsoft

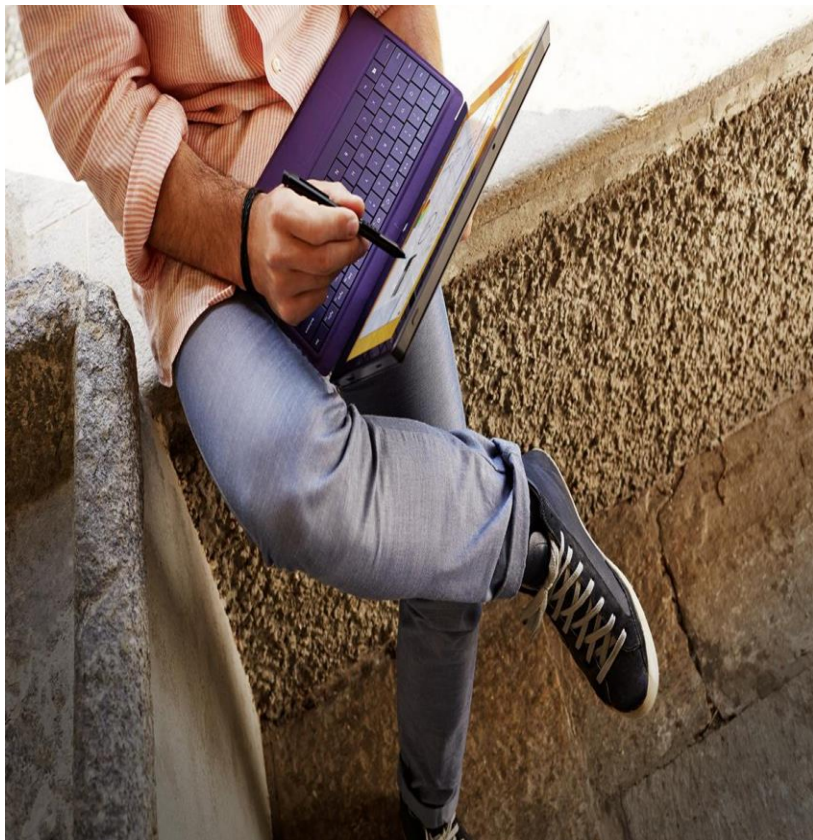
- Solutions can now be created and managed from the Power Apps maker portal (<https://make.powerapps.com>)
- Once you login, select the instance you need from the *Environment* drop-down in the upper right-hand area of the portal screen
- Choose an appropriate action from the command bar or select *Solutions* from the left panel and a specific solution from the list with which you want to work



Building Solutions



Building Solutions – Pre-development Checklist



- ☐ Use the appropriate *Publisher* created specifically for your solutions
- ☐ Follow your defined Segmentation governance when choosing which components you will modify
- ☐ Select/Add **only the components necessary** for your development to your new solution
- ☐ Avoid including ALL components for an entity unless the entity does not already exist in your target instance(s)



Building Solutions – Simple Segmentation Example

Vijay (Web Developer/Customizer)	Carol (Web Developer/Customizer)	Ralph (C# Developer)	T'mika (System Admin)
<ul style="list-style-type: none">• Sales<ul style="list-style-type: none">• Entities<ul style="list-style-type: none">• Account<ul style="list-style-type: none">• Fields (Attributes)• Forms• Views• Contact (Shared)• Lead• Opportunity• ...• Dashboards• Sales Reports• App Modules• Sales Flows• Sales Custom Controls	<ul style="list-style-type: none">• Service<ul style="list-style-type: none">• Entities<ul style="list-style-type: none">• Incident• Contacts (Shared)• Queues• ...• Dashboards• Service Reports• App Modules• Service Flows• Service Custom Controls	<ul style="list-style-type: none">• Plugins• Connectors• Logic App• Virtual Entities• Integration• Generic Custom Controls• ...	<ul style="list-style-type: none">• Site Map• Command Ribbon• Solution Publishers• Security Model• FLS Profiles• ...

- Segmentation allows for multiple developers to work within the bounds of a single instance and potentially within a single solution, if necessary, as it provides a relatively easy means of avoiding most component conflicts
- Ensure your governance process provides an effective mechanism to resolve cross-entity conflicts in the instance that your Segmentation model doesn't account for a particular entity/component collision
- Development Release Notes are an easy way for your developers to identify the impacted entities and components for each version of the solution they create/update



Building Solutions – Publisher Benefits

- Can be associated with multiple solutions
- Effectively identify which set of components a solution addresses (or should address)
- Defines the customizations prefix (***yourprefixhere***_entityname.attribute) and base option set values
- Important - When importing new assets through a managed solution – the associated publisher “owns” those assets

Best Practice

*Use Publishers to help define how you **segment** solution development to facilitate the management of configuration, customization and other components.*

For more information on Segmentation and Publishers, please reference the following docs:

- <https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm#solution-publisher>
- <https://docs.microsoft.com/en-us/power-platform/alm/segmented-solutions-alm>
- <https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/create-solution#create-a-segmented-solution>



Building Solutions – Segmentation Benefits

- Reduces collisions caused by multiple team members working on assets of a component such as an entity
- Solutions are smaller and less complex which simplifies on-going management
- More compact solutions decrease solution import time
- Reduces the code base stored in your source control.
- Provides easy traceability for potential errors introduced into a solution
- Facilitates rapid development models where time-to-implement is critical



Including New Power Platform Components



New PP Components – Embed Canvas Apps

Embedding a Canvas App within a Model-driven App Form is one of the more popular features available with the new Power Platform.

The screenshot shows a model-driven app form for 'Adventure Works (Sample)'. The top navigation bar includes options like New, Deactivate, Connect, Assign, Email a Link, Delete, Refresh, Process, and Share. The form is divided into several sections: 'ACCOUNT INFORMATION' with fields for Account Name, Phone, Fax, Website, Parent Account, and Ticker Symbol; 'ADDRESS' with fields for Address 1: Street 1 and Address 1: Street 2; 'Timeline' with a note entry field; and 'CONTACTS' which is highlighted with a red box. The 'CONTACTS' section displays a profile card for 'Nancy Anderson (sample)' with the email 'someone_c@example.com' and a row of three contact photos.

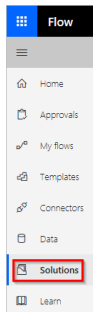
- Embedded Canvas Apps are solution-aware and exist in the solution in which they were created or added (like any other custom control)
- Ensure that you consider your user security model when sharing an embedded canvas app and deploying it in different environments.
- There are important ALM limitations and considerations with this feature, for more information, please see <https://docs.microsoft.com/en-us/powerapps/maker/model-driven-apps/embedded-canvas-app-guidelines>



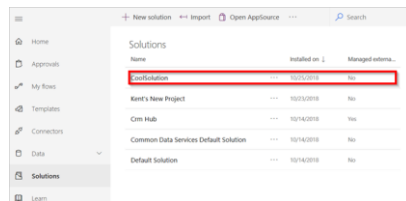
New PP Components - Power Automate

Incorporate Power Automate flows into your existing solutions

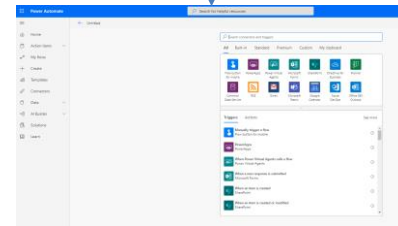
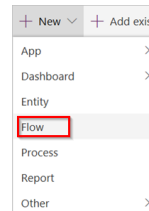
1. Login to Power Automate.
2. Select **Solutions** from the navigation bar



3. Select the solution in which you'll create the flow



4. Select + **New** and then select **Flow**. Power Automate maker then opens



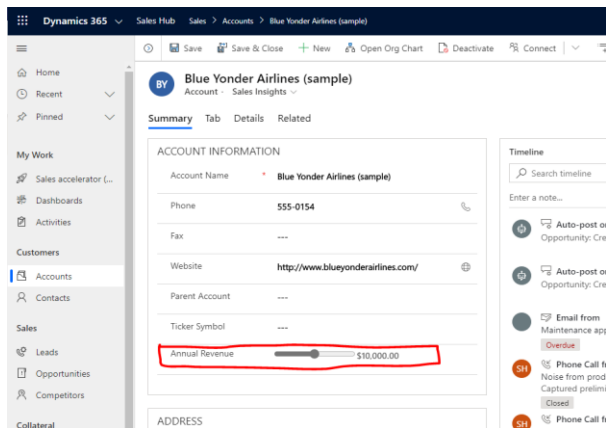
NOTE: *Importing a solution containing a flow will not automatically create or associate required connections. The flow must be edited to fix the connections - If using managed solutions, this creates an active customization in the unmanaged layer. Therefore, subsequent solution updates to the flow will not be reflected.*

For a more detailed description of this and other related Power Automate solution tasks, see <https://docs.microsoft.com/en-us/power-automate/overview-solution-flows>

5. Build your flow with the appropriate connectors and triggers
6. When done, **Save** your flow



New PP Components – Power Apps Component Framework



Best Practice

We recommend that when you establish a segmentation model for your solution(s) you include the most appropriate model for packaging and deploying PCF Controls.

- The Power Apps Component Framework (PCF) allow Canvas App-like controls to be created for use in both Model-driven and Canvas Apps.
- Allows you to build custom controls but still leverage our low-code/no-code methodology.
- For deployment, if you plan to create a generic library of PCF components for your configuration team to leverage, include these in a separate managed solution

For more information on PCF control development see <https://docs.microsoft.com/en-us/powerapps/developer/component-framework/overview>
For more information on PCF solution creating and deployment see <https://docs.microsoft.com/en-us/power-platform/alm/component-framework>



New PP Components - Env Variables

Power Apps

Environment Contoso

+ New + Add existing Delete Export Publish all customizations

Removed components successfully.

Solutions > Contoso

Display name	Name	Type	Managed
Contact	contact	Entity	

New environment variable

Enter information about this variable that will help others use it when it's imported into their environments. [Learn more](#)

Display name *

L1 Approver

Name *

contoso_L1Approver

Description

L1 approver email

Data Type *

Text

Default Value

L1approvers@contoso.com

Current Value

Override the default value by setting the current value for your environment.

+ New value

Save Cancel

Environment Variables (*still in public preview*) allow you transport in-app configuration data or maintain environment-specific configuration data in solutions, eliminating the need to store it in custom entities and migrate it with tools like the configuration migration utility.

These are stored in Solutions as a component leveraging Key-Value Pairs which means they can also be managed with Source Control.

For more information regarding Environment Variables, their usage and deployment options:

- See this blog post: <https://powerapps.microsoft.com/en-us/blog/environment-variables-available-in-preview/> or
- Current Public Preview documentation <https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/environmentvariables>

These variables can be added to or modified directly in the Solution definition.



New PP Components – *Cautionary Guidance*

- ✓ Some of the features covered in this discussion are still in or recently discharged from Preview status.
- ✓ PLEASE ensure that you check the publicly provided documentation on each of these to understand the Current/Known Limitations with each and are prepared to handle accordingly within your implementation
- ✓ Additionally, please note that any feature in still in a state of Preview is NOT SUPPORTED for a Production deployment.



Validating Solutions With Solution Checker



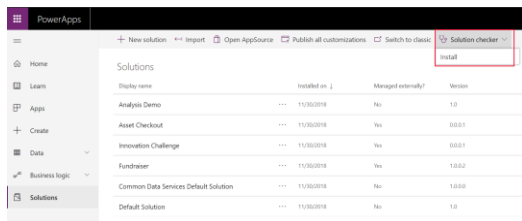
Building Solutions – Solution Checker

Solution Checker provides you the ability to have your solution verified against documented best practices, known deprecations, standard performance challenges, etc.

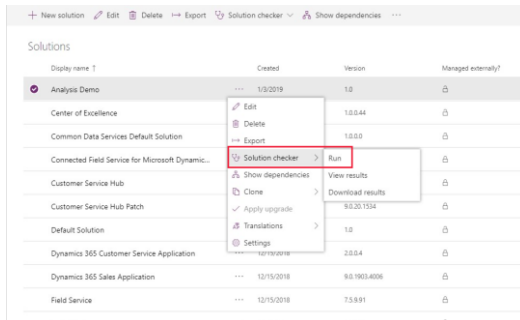
Best Practice

You should run the Solution Checker against every solution you plan to publish and/or distribute.

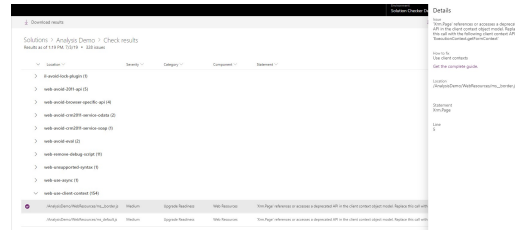
Enable the Checker



Run against your solution



Review and act on the results



Building Solutions - Solution Checker

Environment
Solution Checker De

Download results

Solutions > Analysis Demo > Check results
Results as of 1:19 PM, 7/3/19 • 328 issues

Location

Severity

Category

Component

Statement

> il-avoid-lock-plugin (1)

> web-avoid-2011-api (5)

> web-avoid-browser-specific-api (4)

> web-avoid-crm2011-service-odata (2)

> web-avoid-crm2011-service-soap (1)

> web-avoid-eval (2)

> web-remove-debug-script (11)

> web-unsupported-syntax (1)

> web-use-async (1)

> web-use-client-context (154)

✓	/AnalysisDemo/WebResources/ms_border.js	Medium	Upgrade Readiness	Web Resources	'Xrm.Page' references or accesses a deprecated API in the client context object model. Replace this call with
	/AnalysisDemo/WebResources/ms_default.js	Medium	Upgrade Readiness	Web Resources	'Xrm.Page' references or accesses a deprecated API in the client context object model. Replace this call with

Details

Issue
'Xrm.Page' references or accesses a deprecated API in the client context object model. Replace this call with the following client context API: 'ExecutionContext.getFormContext'

How to fix
Use client contexts

[Get the complete guide.](#)

Location
/AnalysisDemo/WebResources/ms_border.js

Statement
Xrm.Page

Line
5



Validates your solution(s) and provides issue details and guidance



Identifies Best Practice violations, feature/function deprecations, javascript inefficiencies, etc.



Should be run anytime you make changes to a solution before promoting to a managed solution in later environments



Currently works only with model-driven apps



Detailed documentation on this feature can be found [here](#)



Solution Management Tools

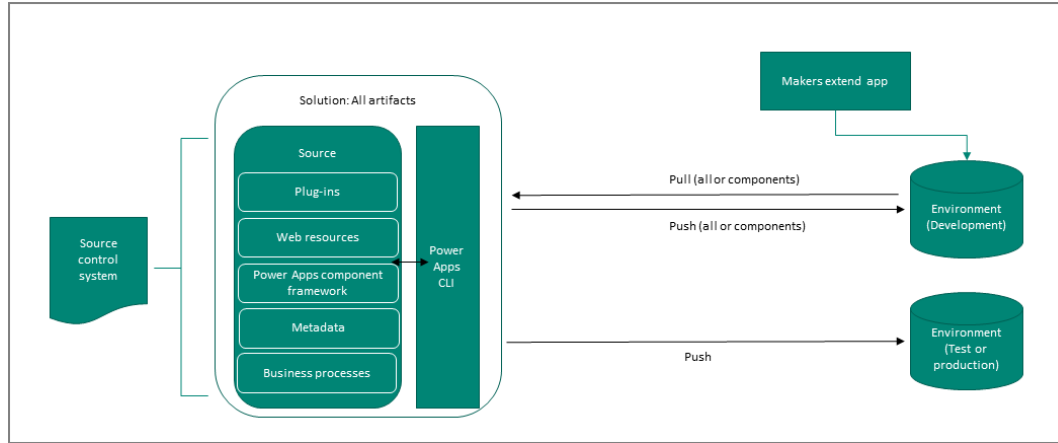


Solution Management Tools - SolutionPackager

- SolutionPackager is a command-line tool that can be used to both decompose and recompose solution files
- Available as part of the *MicrosoftCrmSdk.CoreTools* NuGet package
- Decomposition allows for individual .xml files to be created in a definable structure that can then be integrated into a source control system or other means of safe archival
- Recomposition is primarily aimed at users of source control that have made changes to different components of a solution and are now ready to move the active solution back into the appropriate development instance.



Solution Management Tools – Source Control



- Enables you to have a greater degree of control at a granular level over the different types of components within an MDA solution
- You can leverage a source control system to provide tasks not available OOTB with MDA solutions like branching and merging
- Source control centrality helps you to achieve “Healthy ALM” because the assets maintained in the SCS are the “single source of truth”
- You can use any source control system that supports extension and automation using the Power Apps Command Line Interface (PowerApps CLI), PowerShell or custom code



Solution Management Tools – More Info

- Here are additional documentation links you can read to gain a better understanding of how various tools for Solution Management can be implemented as part of your ALM Strategy and Automation methods:
- <https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/download-tools-nuget>
- <https://docs.microsoft.com/en-us/power-platform/alm/solution-packager-tool>
- <https://docs.microsoft.com/en-us/power-platform/alm/basics-alm#source-control>
- <https://docs.microsoft.com/en-us/power-platform/alm/tools-apps-used-alm#version-control-system>
- <https://docs.microsoft.com/en-us/powerapps/developer/common-data-service/powerapps-cli>
- <https://docs.microsoft.com/en-us/power-platform/alm/use-source-control-solution-files>



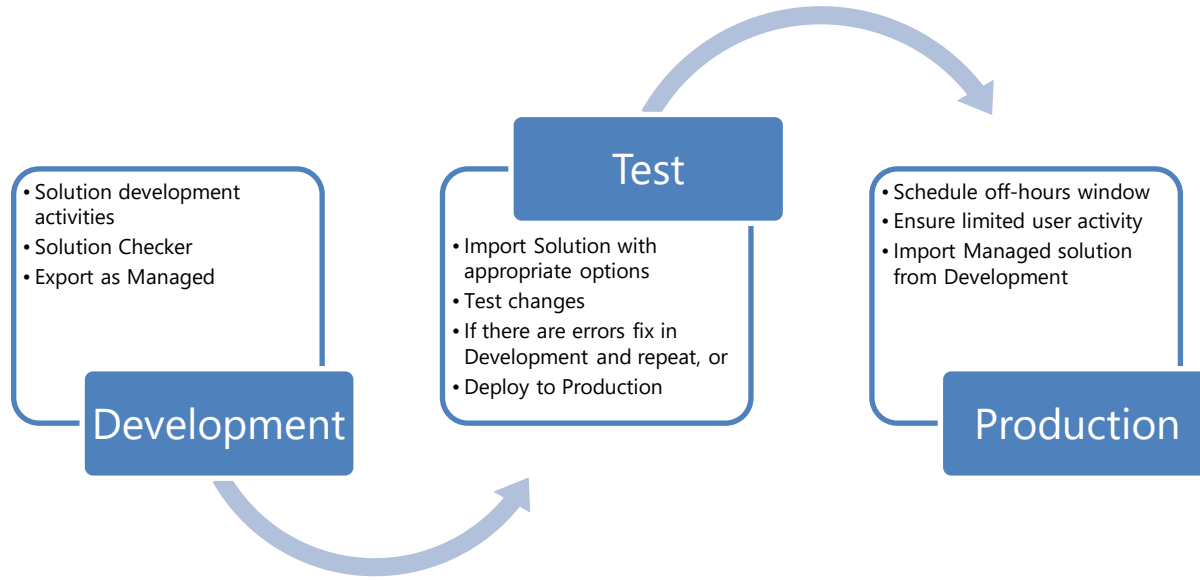
Deploying Solutions



The “Manual” Method



Deploying Manually - Overview



- Simple but should still be managed with appropriate governance
- Opportunities still exist for human error to create issues



Deploying Manually – Exporting Your Solution

The screenshot shows the Power Apps 'Solutions' page. The 'Solutions' table lists various solutions, with 'New Test Solution 1' selected. The 'Export this solution' dialog is open, showing the 'Export as' options. The 'Managed (recommended)' option is selected, and the 'Export' button is highlighted.

Display name	Name	Created	Version	Managed externally?	Solution check
New Test Solution 1	NewTestSolution1	6/9/2020	1.0.0.0		Results as of 10:40 AM, 6/12/20
Contextual Help Base	msdyn_ContextualH...	11/9/2019	1.0.0.10		Checked by Microsoft
Contextual Help	msdyn_ContextualH...	11/9/2019	1.0.0.10		Checked by Microsoft
AppointmentTest	AppointmentTest	8/19/2019	1.0.0.0		Hasn't been run
Power Apps Checker Base	msdyn_PowerAppsC...	7/14/2019	1.2.0.16		Checked by Microsoft
Power Apps Checker	msdyn_PowerAppsC...	5/4/2019	1.2.0.16		Checked by Microsoft
Playbook App	msdyn_playbook_app	4/27/2019	9.0.2005.1010		Checked by Microsoft
LinkedIn Sales Navigator for Microsoft Dynamics	msdyn_LinkedinSale...	4/27/2019	3.0.1.0350		Checked by Microsoft
Sales Navigator for Dynamics 365 Unified Inter...	msdyn_SalesNaviga...	4/27/2019	3.0.1.0350		Checked by Microsoft
LinkedIn - Sales Navigator	LinkedIn	4/27/2019	3.0.0.0630		Checked by publisher
Common Data Services Default Solution	CdSabc	8/26/2018	1.0.0.0		Hasn't been run
Field Service	FieldService_anchor	3/21/2018	7.4.0.80		Checked by Microsoft
Field Service UCI App Module	FieldServiceUCIApp...	3/21/2018	7.4.0.80		Checked by Microsoft
Field Service App Module	FieldServiceAppMod...	3/21/2018	7.4.0.80		Checked by Microsoft
Field Service	FieldService	3/21/2018	7.4.0.80		Checked by Microsoft
Project Service Automation	ProjectService_Ancor	3/21/2018	2.4.0.51		Checked by Microsoft
Project Service Automation App Module	ProjectServiceAppM...	3/21/2018	2.4.0.51		Checked by Microsoft

Export this solution

Version number *
Current version 1.0.0.0
1.0.0.0

Export as

☒ **Managed (recommended)**
The solution is moving to a test or production environment. [Learn more](#)

☐ **Unmanaged**
The solution is moving to another development environment or source control. [Learn more](#)

Export **Cancel**

➤ First step after completing a Solution build in DEV and you're ready for deeper testing (QA/UAT)

➤ Use Power Apps (make.powerapps.com) to manage your solutions

➤ Select the solution you wish to migrate and choose Export in the toolbar

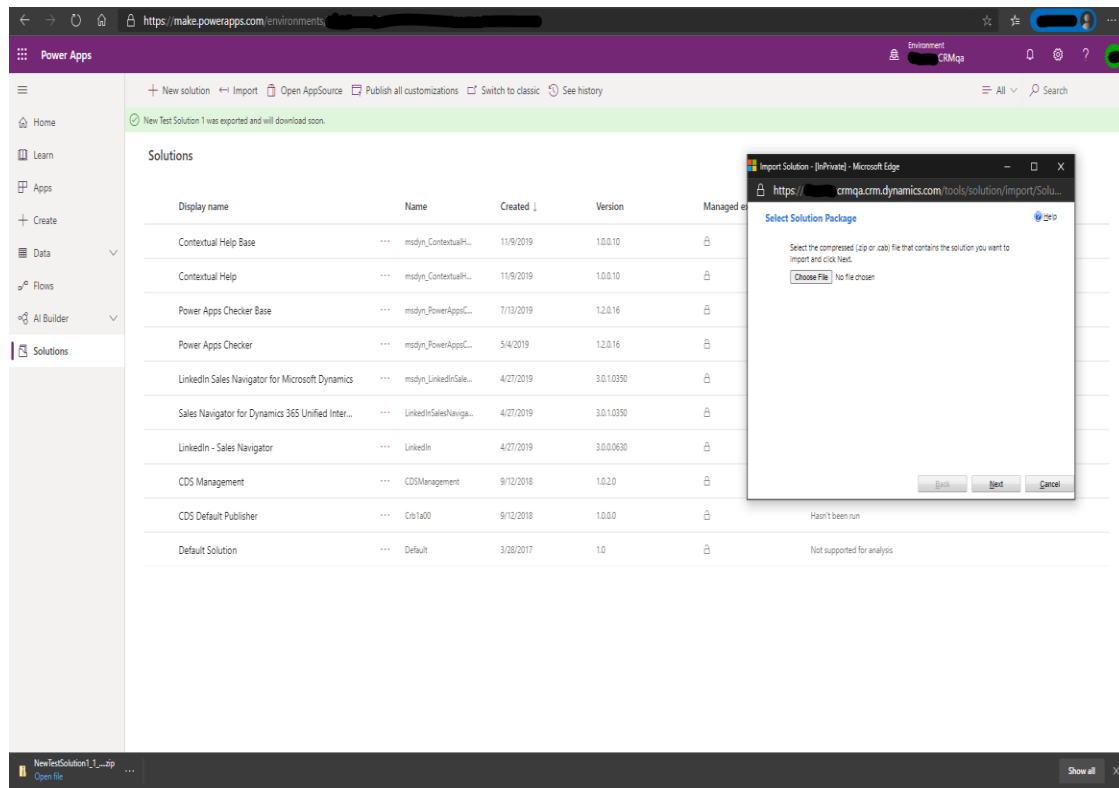
➤ Creates a .zip file that contains all of the Solution XML files

➤ You should extract as a Managed solution

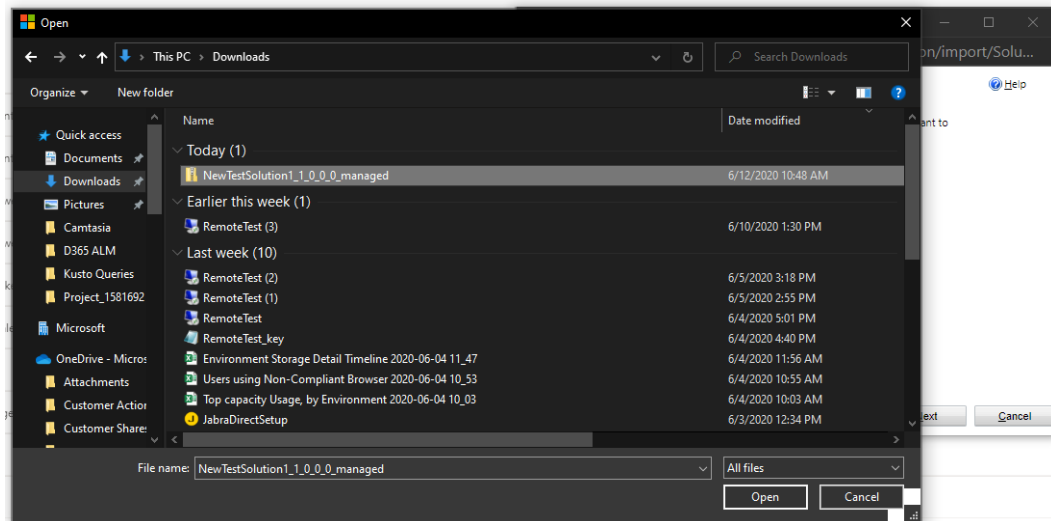


Deploying Manually – Importing Your Solution

- Select the target environment
- Choose Import from the toolbar
- Press the Choose File button and select your local .zip solution file you exported from your DEV environment
- Click Next

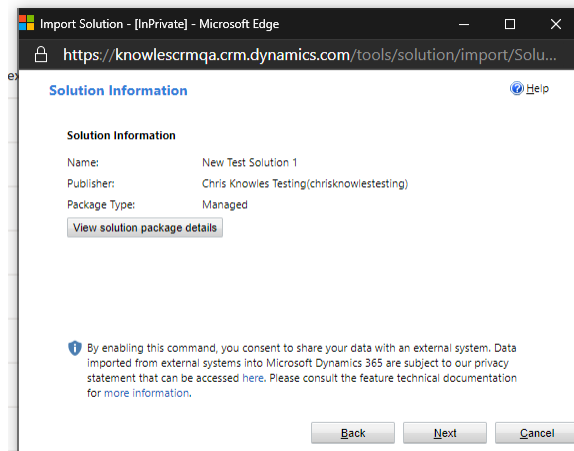


Deploying Manually – Importing Your Solution

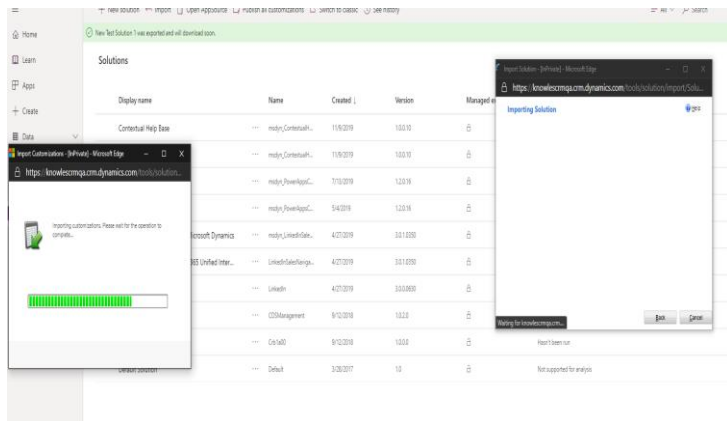


Select your solution .zip file and press the Open button

✓ If the correct file was chosen and resembles the screen shown to the right looks, press *Next*



Deploying Manually – Importing Your Solution



Import Options

This solution package contains an update for a solution that is already installed.

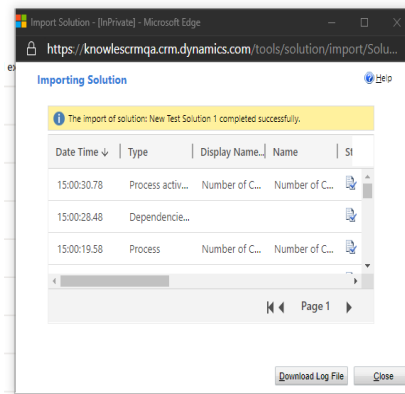
Solution Action (Learn more)

- Upgrade (recommended)
This option upgrades your solution to the latest version and rolls up all previous patches in one step. Any components associated to the previous solution version that are not in the newer solution version will be deleted.
- Stage for Upgrade
This option upgrades your solution to the higher version, but defers the deletion of the previous version and any related patches until you apply a solution upgrade later.
- Update (not recommended)
This option replaces your solution with this version. Components that are not in the newer solution will not be deleted and will remain in the system.

Previous customizations on components included in this solution (Learn more)

- Maintain customizations (recommended)
This option maintains any unmanaged customizations performed on components, but also implies that some of the updates included in this solution will not take effect.
- Overwrite customizations (not recommended)
This option overwrites or removes any unmanaged customizations previously performed on components in this solution. This option does not affect components that support merge behavior (forms, sitemap, ribbon, app modules). Components that have other managed solutions on top of the existing solution you are replacing do also still remain on top and are not affected by this option.

Back Import Cancel



Hasn't been run

- If the solution is new to the target environment, your import will begin
- If the solution already exists in your target instance you will see a window of options for how you want the import to behave.
- Select the appropriate option(s) and choose *Import* and your import will begin
- The solution is then imported into the target environment and progress is reported
- When it completes or if there are errors, you will have the opportunity to capture the log file so you can see the details of the failing process and where the error occurred
- If the import was successful, make sure to select publish customizations in the target environment
- Validate that your changes are active and working in the target instance
- If everything looks good, you can now deploy the same solution into any other target instance (like Production)

For more information on importing solutions, see <https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/import-update-export-solutions>

For more information on updating/upgrading solutions, see <https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/update-solutions>

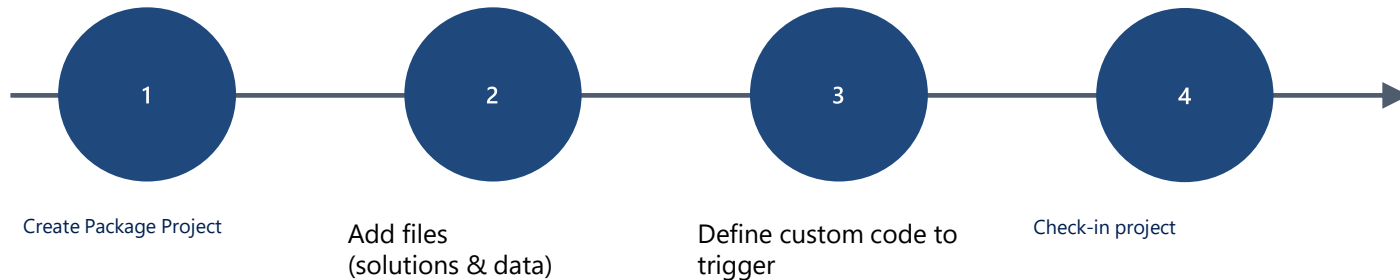


Package Deployer



Package Deployer

- Find more information on the Package Deployer tool [here](#)
- Assists with deploying applications comprising multiple solution and configuration data
- A [Visual Studio project template](#) is provided to construct package definitions
- Project can be version controlled and used in build process

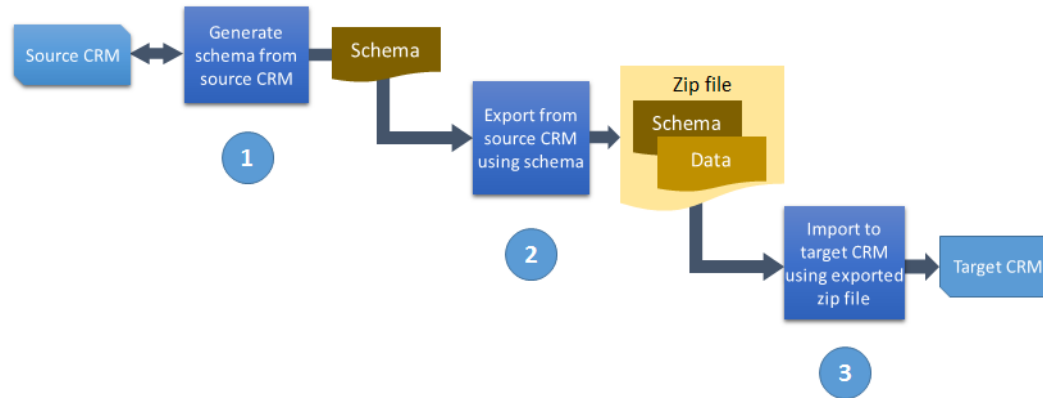


Configuration Migration Tool



Configuration Migration Tool

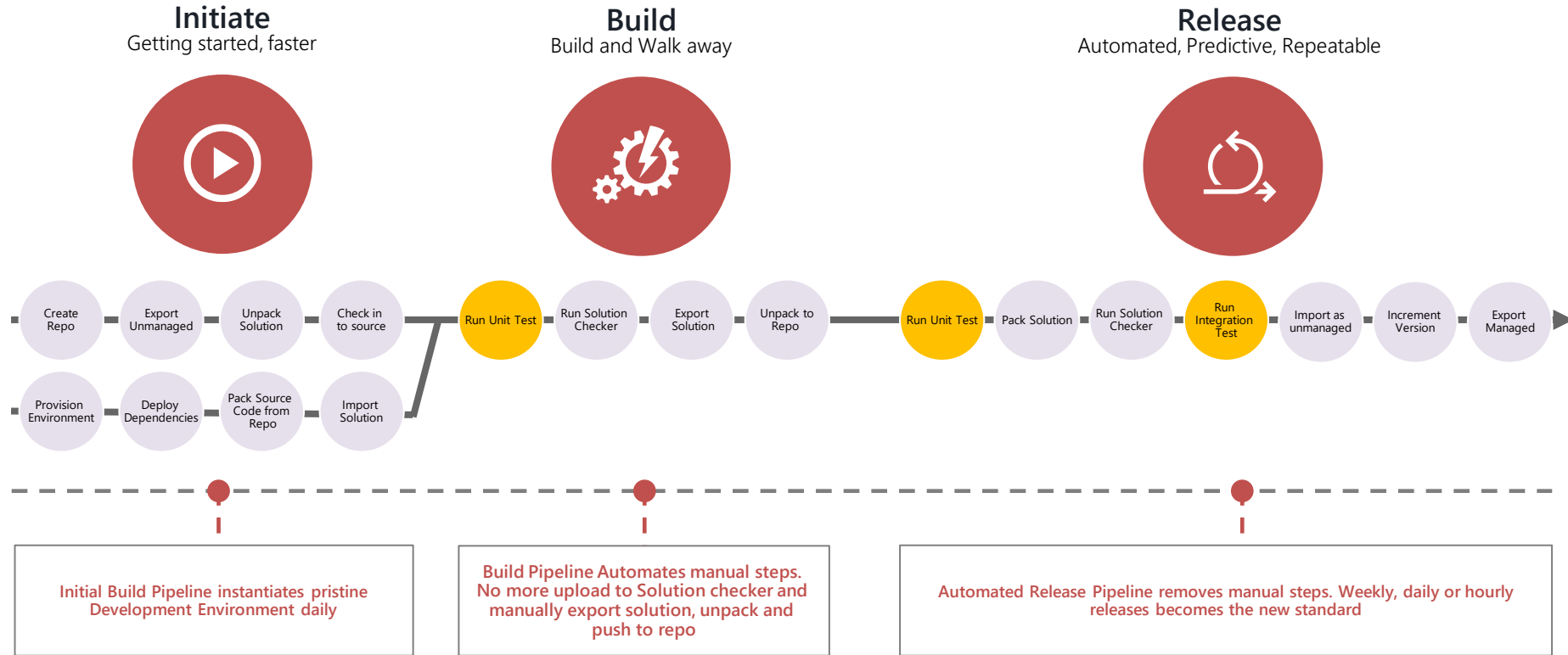
- Designed to migrate custom functionality configuration data from one environment to another.
- The configuration data defined in the Unified Service Desk application for Dynamics 365 is a good example of the entities that would have to be migrated using the Configuration Tool
- Not designed to be a user data migration tool.



Tools for Automation



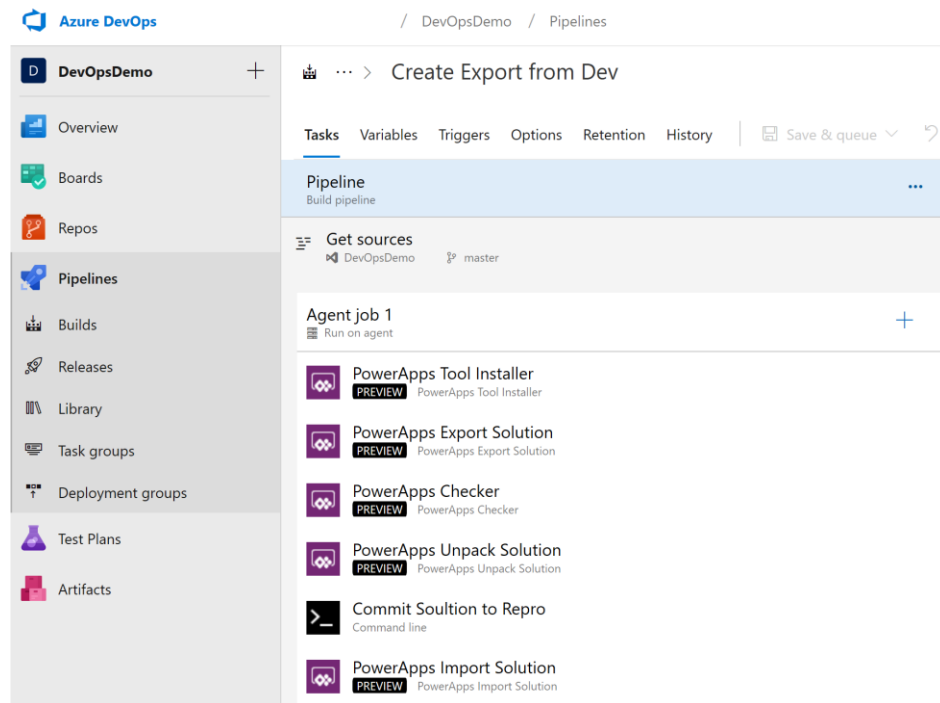
Tools For Automation - Azure DevOps



Tools for Automation – PowerApps Build Tools

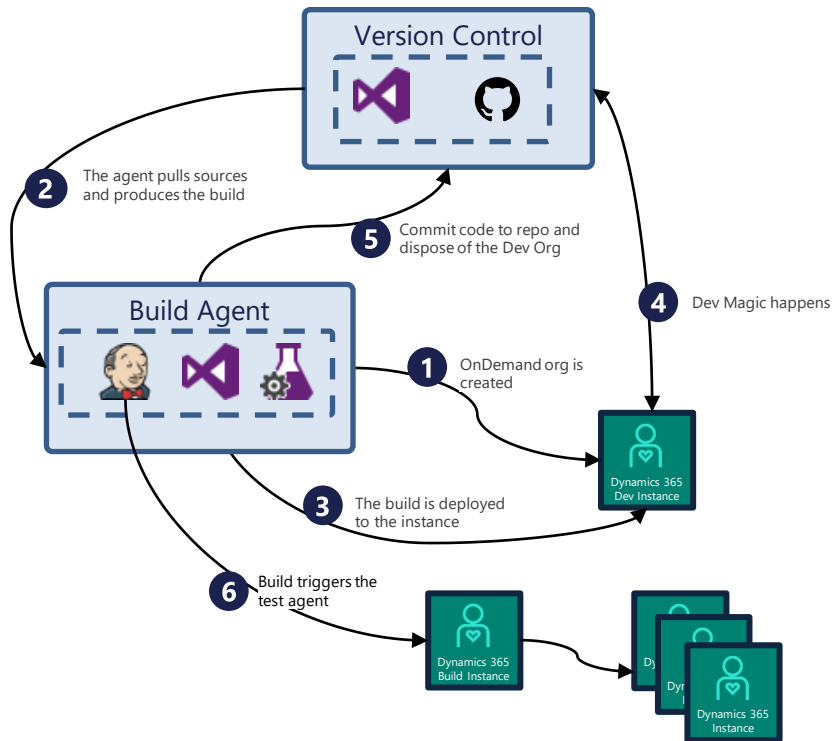
Use PowerApps Build Tools to automate common build and deployment tasks related to PowerApps.

- ✓ Synchronization of solution metadata (a.k.a. solutions) between development environments and source control
- ✓ Generating build artifacts
- ✓ Deploying to downstream environments
- ✓ Provisioning/de-provisioning of environments
- ✓ Ability to perform static analysis checks against your solution using the PowerApps checker service.



Provisioning CI Instances

Used to deploy solutions to development environments



A build is queued and initiates:

- 1 Org creation
- 2 The build agent pulls sources from version control and produces the build
- 3 The solution or package is deployed to the instance
- 4 Dev Magic happens
- 5 Solution checker runs and the code/configuration is pushed to source control and the instance destroyed
- 6 Run automated tests and deploy



Build Process

1. Extract from Dev, store in Source Control and import into Build org

The screenshot shows the 'Create Export from Dev' pipeline configuration. The 'Pipeline' tab is selected, showing a 'Build pipeline' named 'Build pipeline'. The 'Get sources' task is configured with 'DevOpsDemo' as the provider and 'master' as the branch. The 'Agent job 1' is configured to 'Run on agent'. The job contains the following tasks: 'PowerApps Tool Installer' (PowerApps Tool Installer), 'PowerApps Export Solution' (PowerApps Export Solution), 'PowerApps Checker' (PowerApps Checker), 'PowerApps Unpack Solution' (PowerApps Unpack Solution), 'Commit Solution to Repo' (Command line), and 'PowerApps Import Solution' (PowerApps Import Solution).

2. Extract from build org and create Managed Solution

The screenshot shows the 'Build Managed Solution' pipeline configuration. The 'Pipeline' tab is selected, showing a 'Build pipeline' named 'Build pipeline'. The 'Get sources' task is configured with 'DevOpsDemo' as the provider and 'master' as the branch. The 'Agent job 1' is configured to 'Run on agent'. The job contains the following tasks: 'PowerApps Tool Installer' (PowerApps Tool Installer), 'PowerApps Pack Solution' (PowerApps Pack Solution), 'PowerApps Export Solution' (PowerApps Export Solution), and 'Publish Pipeline Artifact' (Publish pipeline artifact).

3. Import managed solution in the upstream orgs

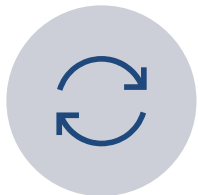
The screenshot shows the 'New release pipeline' configuration. The 'All pipelines' tab is selected, showing a 'New release pipeline' named 'Prod'. The 'Pipeline' tab is selected, showing a 'Deployment process' named 'Prod'. The 'Agent job' is configured to 'Run on agent'. The job contains the following tasks: 'PowerApps Tool Installer' (PowerApps Tool Installer) and 'PowerApps Import Solution' (PowerApps Import Solution).



Test Automation with EasyRepro



UI Automation Test Library for MDA - EasyRepro



TEST AUTOMATION
PROVIDES SIGNIFICANT
ROI



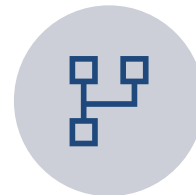
INCREASINGLY
IMPORTANT FOR "ONE
VERSION"



ABSTRACTS TEST CASE
IMPLEMENTATION
FROM UI CONTROL
IMPLEMENTATION



INTUITIVE AND
EFFECTIVE TO GET UP
AND RUNNING



CAN BE EXECUTED AS
PART OF DEVOPS
PIPELINE



UI Automation Test Library for MDA - EasyRepro

- Based on industry-standard Unit Testing models
- Supports various testing needs including Smoke, Regressing, Load, etc.
- Requires time investment for initial learning and setup so start planning now!

Test Name: UCITestOpenActiveAccount

Test Outcome:  Passed

Standard Output

Debug Trace:

Login Command
OnlineLogin.cs>Login Command
BrowserCommand-Execute command
BrowserCommand-Execute command
OpenApp Command
BrowserCommand-Execute command
BrowserCommand-Execute command
OpenSubArea Command
BrowserCommand-Execute command
BrowserCommand-Execute command
BrowserCommand-Execute command
BrowserCommand-Execute command
BrowserCommand-Execute command
OpenRecord Command
BrowserCommand-Execute command

Blog Post Series – <https://community.dynamics.com/crm/b/crminthefield/posts/test-automation-and-easyrepro-01---overview-and-getting-started>



Build/Deploy Automation Demo



Conclusion



Deploying Solutions – Best Practice

Significant challenges created by using and deploying only Unmanaged Solutions:

- *Inconsistent development governance*
- *No ability to successfully implement sustainable Source Control*
- *Customer attempts at Automation/CI were complex and ultimately unsuccessful*
- *Significant Customer, Partner and Microsoft effort-hours expended to resolve dependency conflicts or errant code deployed in Dynamics 365 Online instances*

Best Practice

*Microsoft recommends
Build Unmanaged, Deploy Managed
as the best practice model for managing Solutions with all
Dynamics 365 Legacy and Model-driven Apps.*



Key Takeaways



Dynamics Development is moving from instance centric to **source control centric** providing you with greater control over your development processes



Unmanaged Solutions are for **development only**. Only Managed Solutions are deployed to non-dev orgs



Don't wait to automate! Build **automation** starts **before** any **customizations** or code is developed



Bill's back ... with a question!



Next Steps



Next Steps – Solution Health Check



Build Automation in place

100% automated ALM?



Managed Solutions

Shipping only Managed Solutions outside of Dev environments



Is your ALM process working?

Is your Source Control definitive source of truth?

Are your Dev environments disposable?

Developer isolation?

Service production while working on v-next?



Is your solution created correctly?

Supported techniques only?

Testing in UC and using AppModules?

Single publisher?

Solution Checker?

Separate environments for each solution?

Only have your managed dependencies installed?



Next Steps – Read our ALM Documentation

The screenshot shows the Microsoft documentation page for Application Lifecycle Management (ALM) with the Power Platform. The browser address bar shows the URL <https://docs.microsoft.com/en-us/power-platform/alm/>. The page title is "Application lifecycle management (ALM) with Microsoft Power Platform". Below the title is a subtitle: "Learn how to implement application lifecycle management (ALM) using Microsoft Power Platform." The left sidebar contains a navigation menu with the following items: Overview, ALM basics, Environment strategy, Solutions, Tools and apps used, Implementing healthy ALM, and ALM for developers. The main content area is divided into three columns. The first column, "About ALM with Power Platform", includes sections for OVERVIEW (What is ALM?), CONCEPT (ALM basics with Power Platform, Environment strategy, Tools and apps used in ALM), and For developers (CONCEPT, ALM for developers, Microsoft Power Platform Build Tools for Azure DevOps, Work with solution components, Solution and packaging tools, Manage solutions using custom code). The second column, "Understand and work with solutions", includes sections for CONCEPT (Solution concepts, Understand solution layers, Use solutions to customize, Segment solutions) and Related docs (OVERVIEW, Create apps in Power Apps, What is Common Data Service?, Work with solutions, Power Platform admin guide, Power Platform Center of Excellence (CoE) Kit). The third column, "Implement healthy ALM", includes a CONCEPT section with six scenarios: Scenario 0: ALM for a new project, Scenario 1: Citizen development ALM, Scenario 2: Moving from a single environment, Scenario 3: Moving from unmanaged to managed solutions, Scenario 4: Use DevOps for automation, Scenario 5: Support team development, and Scenario 6: Embrace citizen developers. At the bottom left of the page, there is a "Download PDF" button.

- Located at <https://docs.microsoft.com/en-us/power-platform/alm/>
- Review the Implement healthy ALM scenarios appropriate for your organization.
- **Implement Healthy ALM!**



Thank You!!

- Review the other Tech Talks available to you
- Where to go for help/questions:
 - Docs <https://docs.microsoft.com/en-us/power-platform/alm/>
 - Learning <https://docs.microsoft.com/en-us/learn/dynamics365/>
 - Communities <https://community.dynamics.com/b/>
 - Account Team or Reseller
 - Premier Support
 - If assigned, FastTrack Solution Architect





© 2016 Microsoft Corporation. All rights reserved. Microsoft, Microsoft Dynamics, Office 365, Windows, and other product names are or may be registered trademarks and/or trademarks in the U.S. and/or other countries. The information herein is for informational purposes only and represents the current view of Microsoft Corporation as of the date of this presentation. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information provided after the date of this presentation. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS PRESENTATION.