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

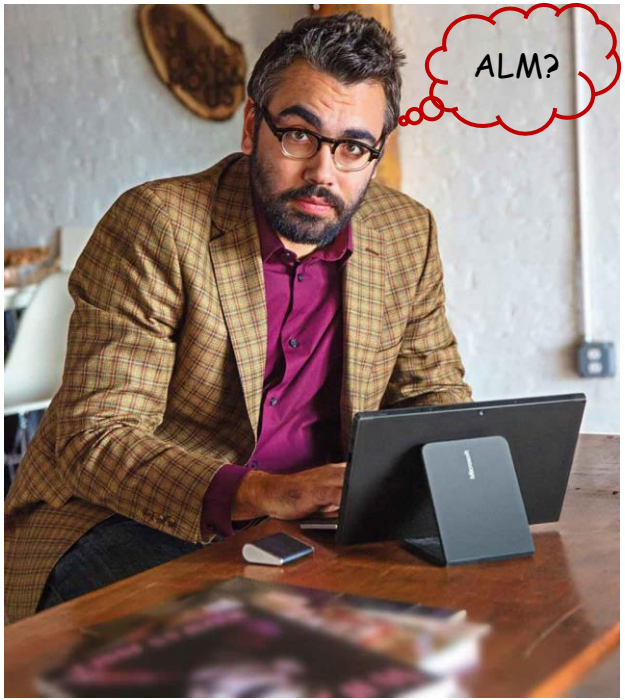
- 1 hour

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

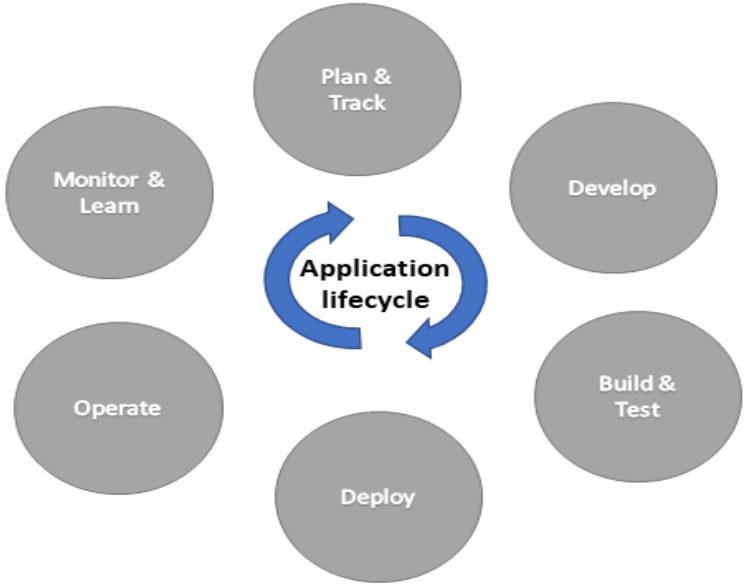
Overview – Application Lifecycle Management



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

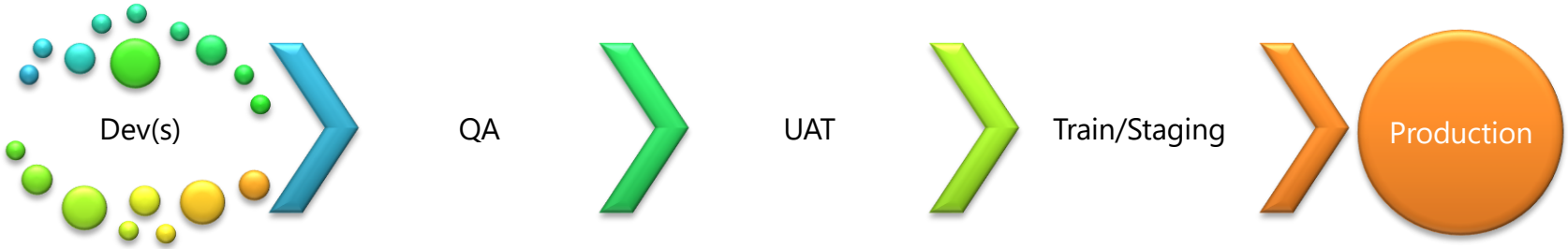
Best Practices

- Strategy and Governance
- Solution Management
- Deploying Solutions
- Build Automation
- Source Control

For detailed documentation on the ALM processes available for the Power Platform, please see https://docs.microsoft.com/en-us/power-platform/alm/

Environment/Storage Planning & Management

Environment Planning – Instance Strategy

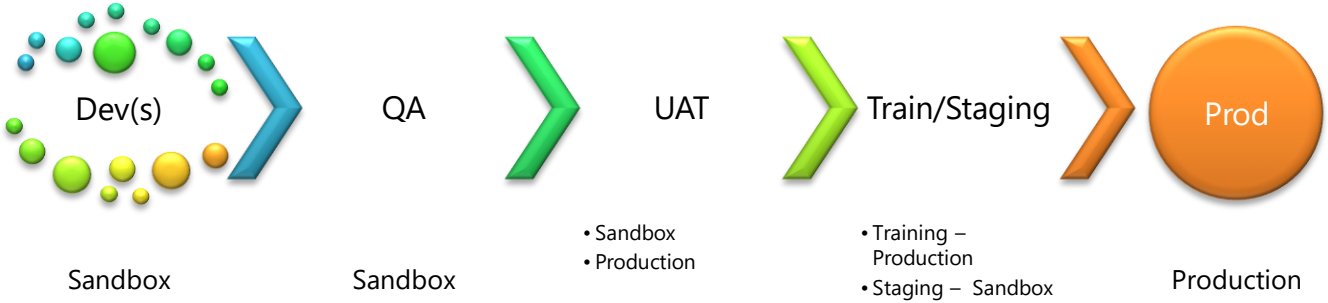


- ✓ Define the instance requirements necessary to support the ideation, build and operation your business
- ✓ Should facilitate project decisions such as the development model, go-live approach, where to conduct a POC (Proof of Concept), UAT (User Acceptance Testing), Hot Fix/Break Fix, etc., identifying 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 centricity**, eliminates dependences on specific instances and allows you to validate your deployment process each time you deploy

For more information see the following blog post https://powerapps.microsoft.com/en-us/blog/establishing-an-environment-strategy-for-microsoft-power-platform/

Environment Planning – Instance Types

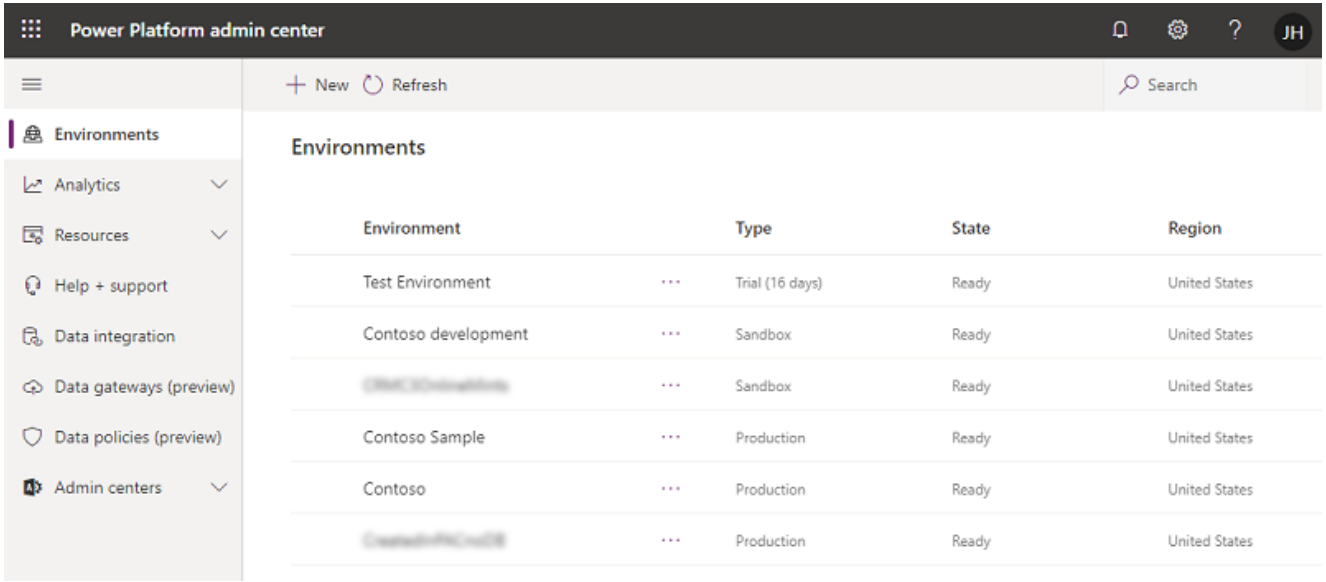
Sandbox	Production
• Development/Test • Lower resource budget • No performance expectations	• Commercial Use • Greater resource budget • Dynamic scalability



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

Environment Planning – Instance Management

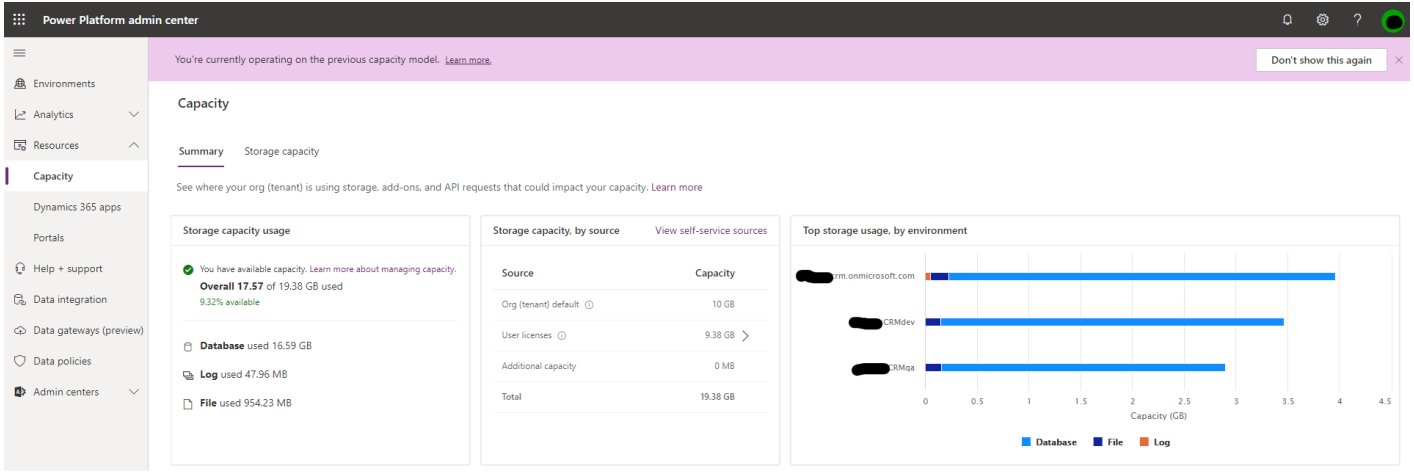
- Instances are CDS environments created to be used with Dynamics 365 Model-driven Apps
- Can be created, configured and managed from the Power Platform Admin Portal.



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

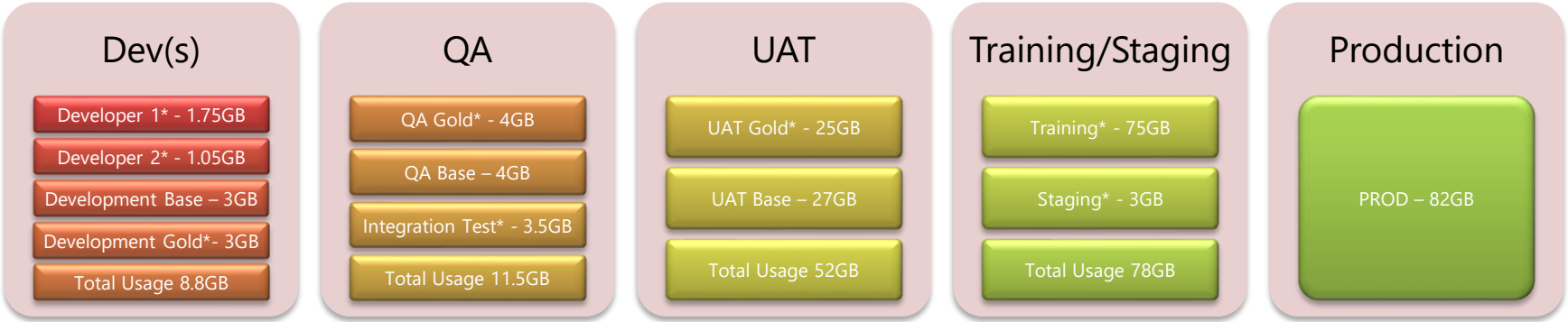
Storage Planning – Determining Needs

- 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



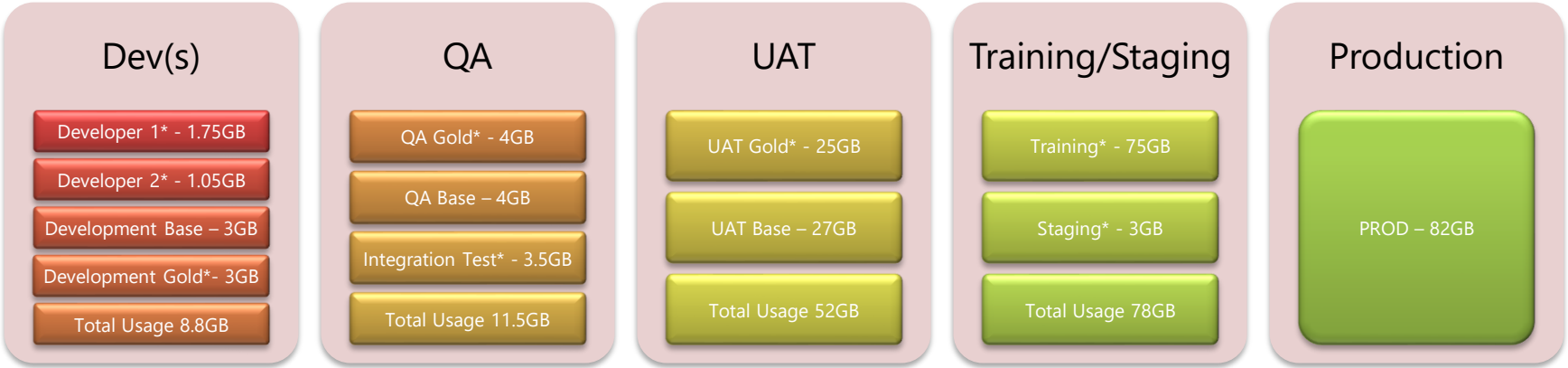
To understand more about storage management and how to properly administrate it, see https://docs.microsoft.com/en-us/power-platform/admin/whats-new-storage

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 unforseen 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

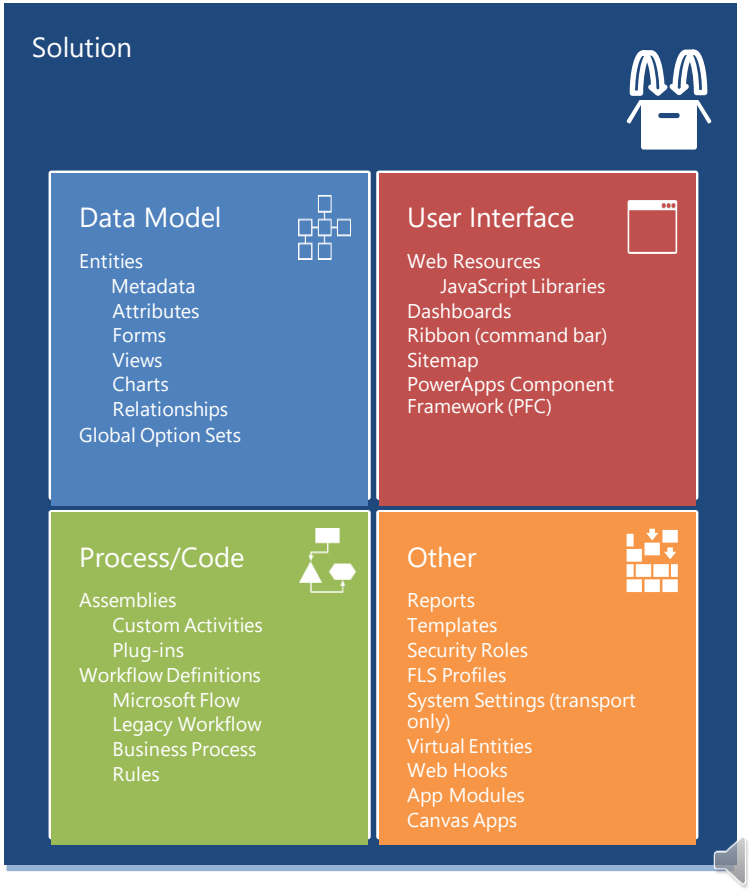
Solution Management

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

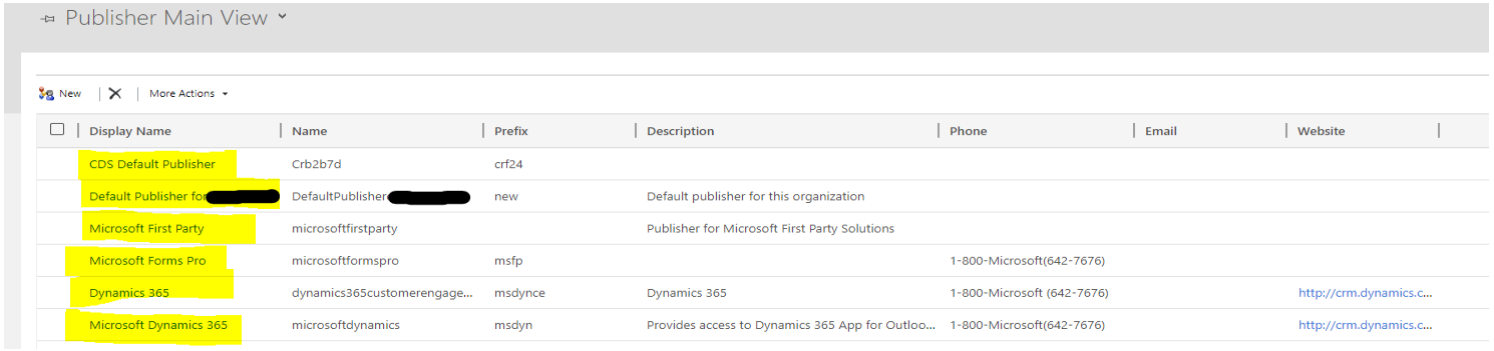


Solution Basics – Solution Types

Unmanaged	Managed
A logical container of customizations existing on top of the system and managed solutions	A "version complete" unmanaged solution that has been exported as managed
Unmanaged solutions are developed	**Managed solutions are deployed**
• 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 *Default* solution is the highest unmanaged solution layer. • NOTE: We do not recommend you develop in or deploy the *Default* solution	• 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

Please read the additional information on Managed and Unmanaged solution types at https://docs.microsoft.com/en-us/power-platform/alm/solution-concepts-alm

Solution Basics - Publishers

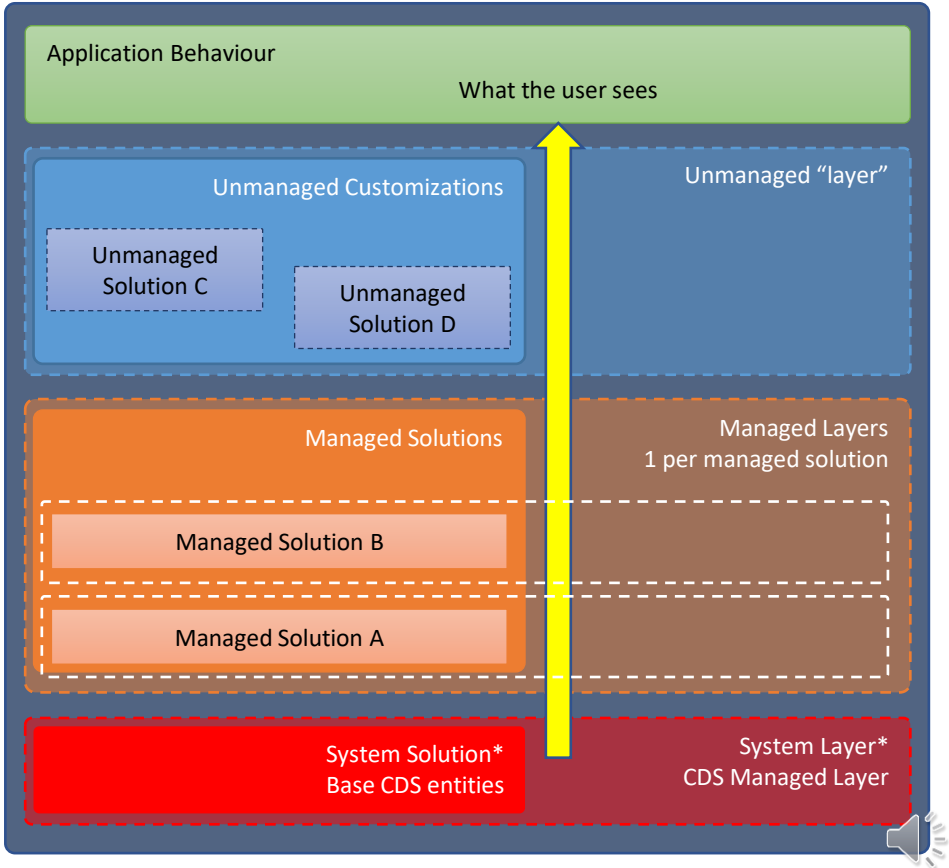


- A Solution-related entity that represents the "owner" of a published solution and the assets contained within.
- You can see your publishers in your instance within UCI by selecting the *Settings Wheel – Advanced Settings*, then under the *Settings* categories, choose *Customization* and then *Publishers*
- 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 systemuser 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

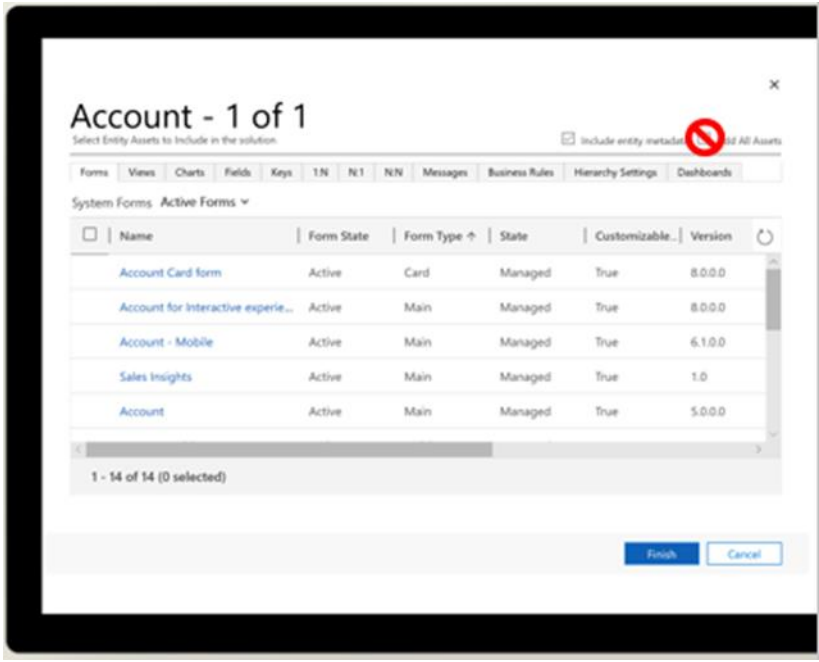
- ✓ Layers describe the dependency chain of a component from the root solution introducing it, through each solution that extends or changes the component's behavior
- ✓ Layers are created through extension of an existing component (taking a dependency on it) or through creation of a new component or 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



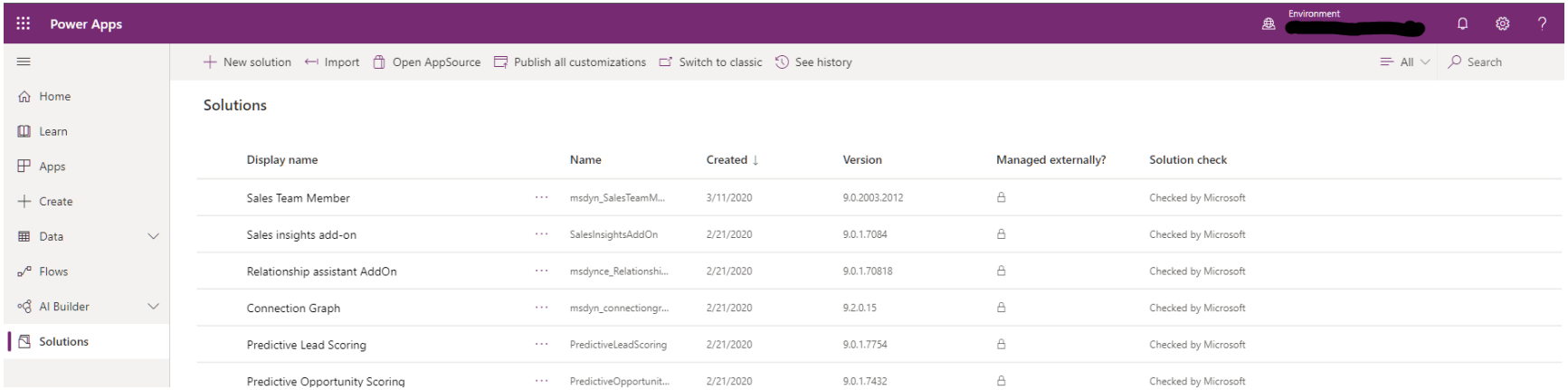
Solution Basics - Segmentation

- ✓ 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**



For more information, please see https://docs.microsoft.com/en-us/power-platform/alm/segmented-solutions-alm

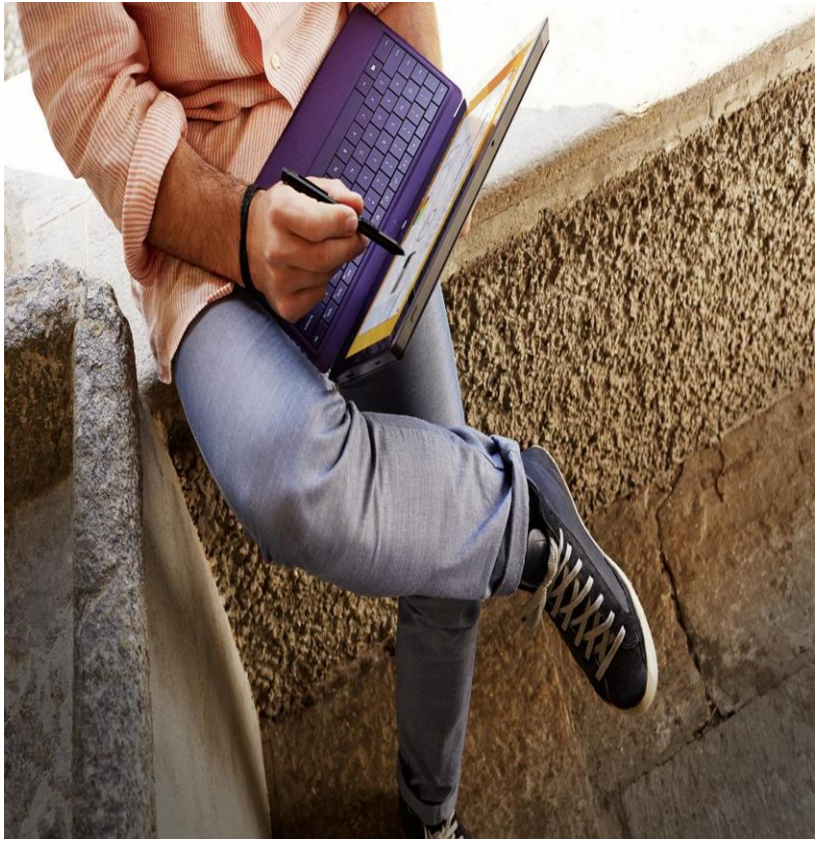
Solution Basics – Manage with Power Apps



- 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)

- Sales
  - Entities
    - Account
      - Fields (Attributes)
      - Forms
      - Views
    - Contact (Shared)
    - Lead
    - Opportunity
    - …
  - Dashboards
  - Sales Reports
  - App Modules
  - Sales Flows
  - Sales Custom Controls

Carol (Web Developer/Customizer)

- Service
  - Entities
    - Incident
    - Contacts (Shared)
    - Queues
    - …
    - Dashboards
    - Service Reports
    - App Modules
    - Service Flows
    - Service Custom Controls

Ralph (C# Developer)

- Plugins
- Connectors
- Logic App
- Virtual Entities
- Integration
- Generic Custom Controls
- …

T'mika (System Admin)

- 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

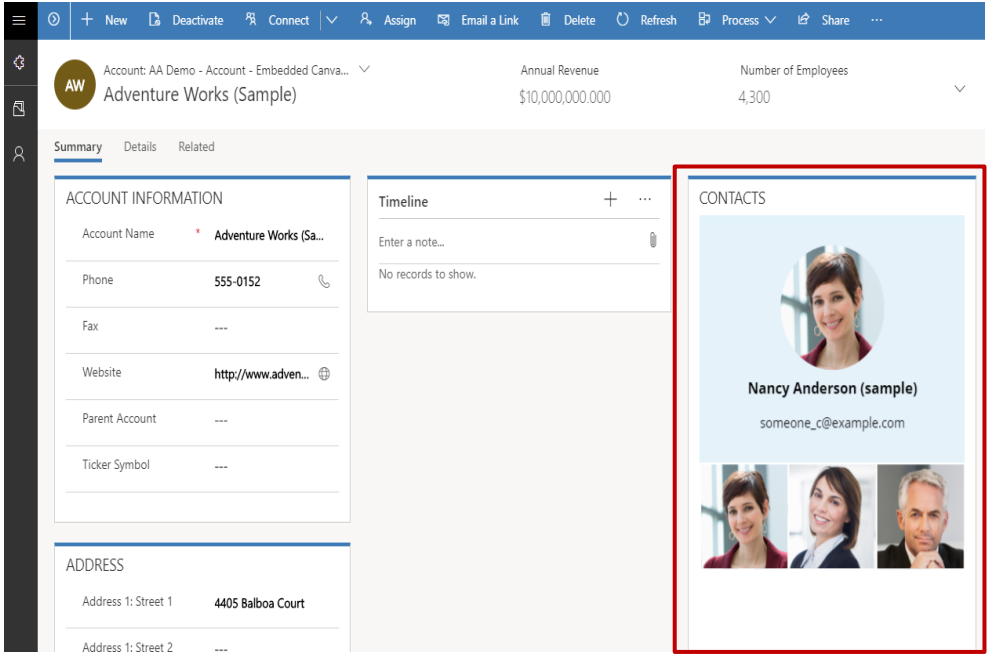
Building Solutions – Segmentation Benefits

- 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.

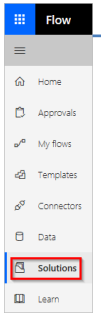


- Embedded Canvas Apps are solution-aware and exist in the solution in which they were created or added (like any other custom control)
- Ensure that you consider your user security model when sharing an embedded canvas app and deploying it in different environments.
- There are important ALM limitations 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

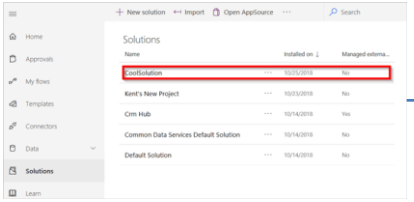
New PP Components - Power Automate

Incorporate Power Automate flows into your existing solutions

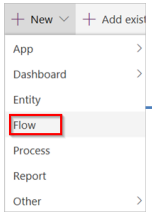
1. Login to Power Automate.
2. Select **Solutions** from the navigation bar

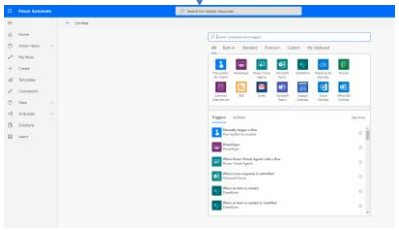


3. Select the solution in which you'll create the flow



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



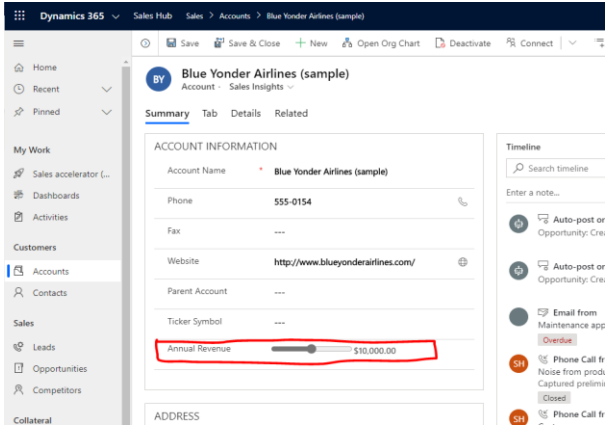


5. Build your flow with the appropriate connectors and triggers
6. When done, **Save** your flow

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

New PP Components – Power Apps Component Framework



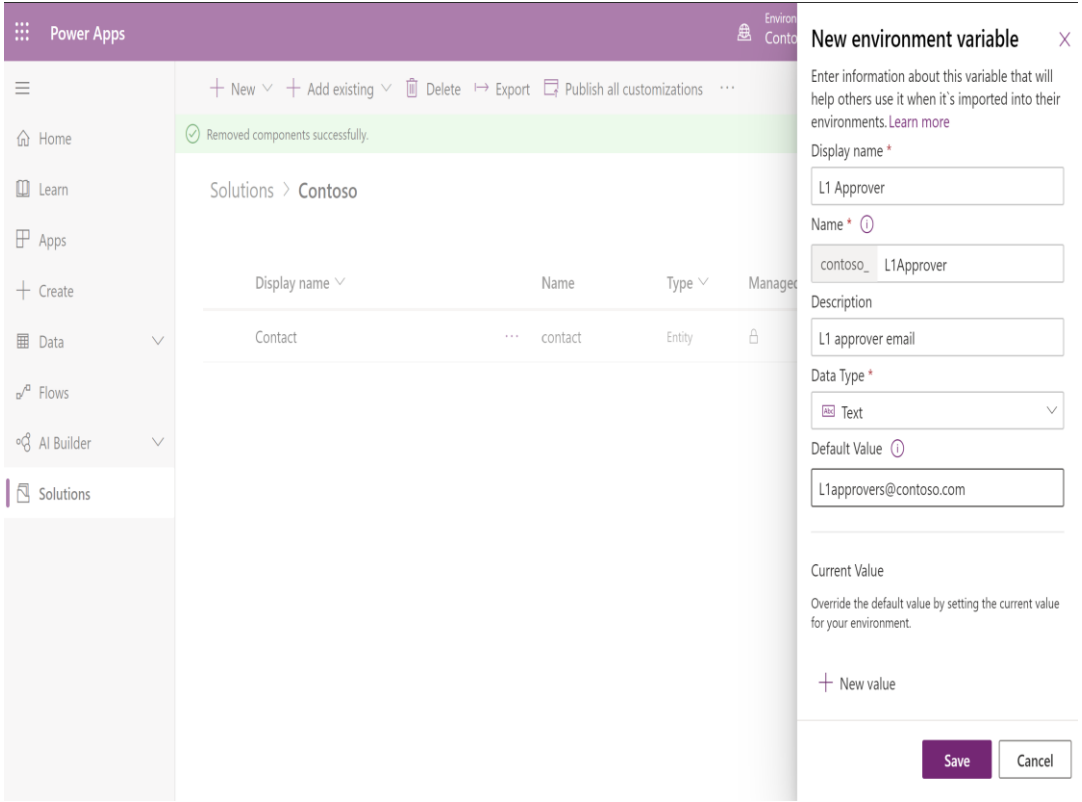
- The Power Apps Component Framework (PCF) allow Canvas App-like controls to be created for use in both Model-driven and Canvas Apps.
- Allows you to build custom controls but still leverage our low-code/no-code methodology.
- For deployment, if you plan to create a generic library of PCF components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

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.

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



- 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*

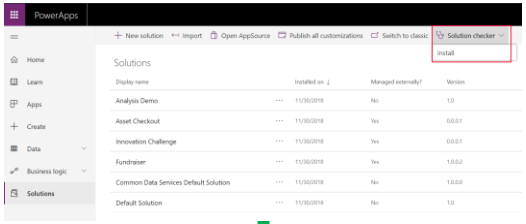
- ✓ Some of the features covered in this discussion are still in or recently discharged from Preview status.
- ✓ PLEASE ensure that you check the publicly provided documentation on each of these to understand the Current/Known Limitations with each and are prepared to handle accordingly within your implementation
- ✓ 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.

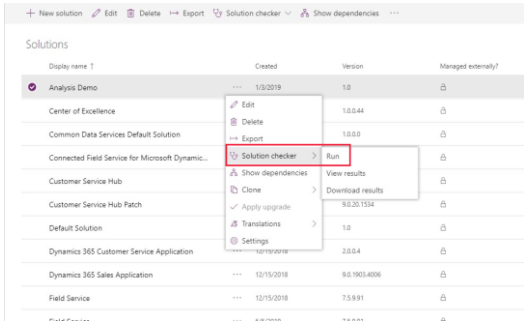
Enable the Checker



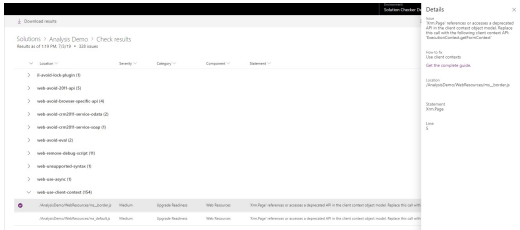
Best Practice

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

Run against your solution

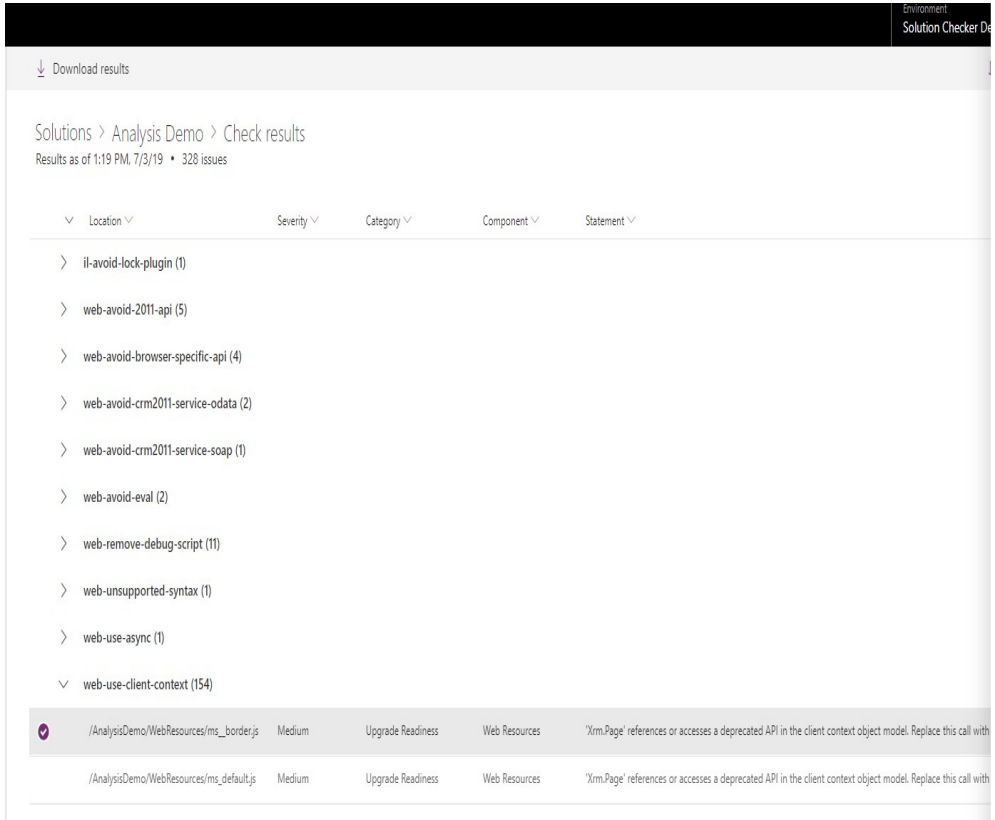


Review and act on the results



For more information, see https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/use-powerapps-checker

Building Solutions - Solution Checker



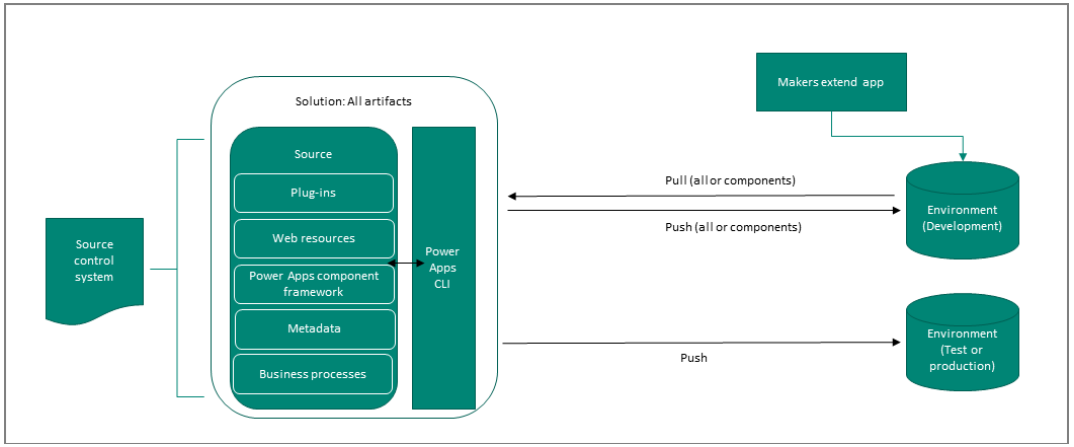
- Validates your solution(s) and provides issue details and guidance
- Identifies Best Practice violations, feature/function deprecations, javascript inefficiencies, etc.
- Should be run anytime you make changes to a solution before promoting to a managed solution in later environments
- Currently works 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 centricity 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

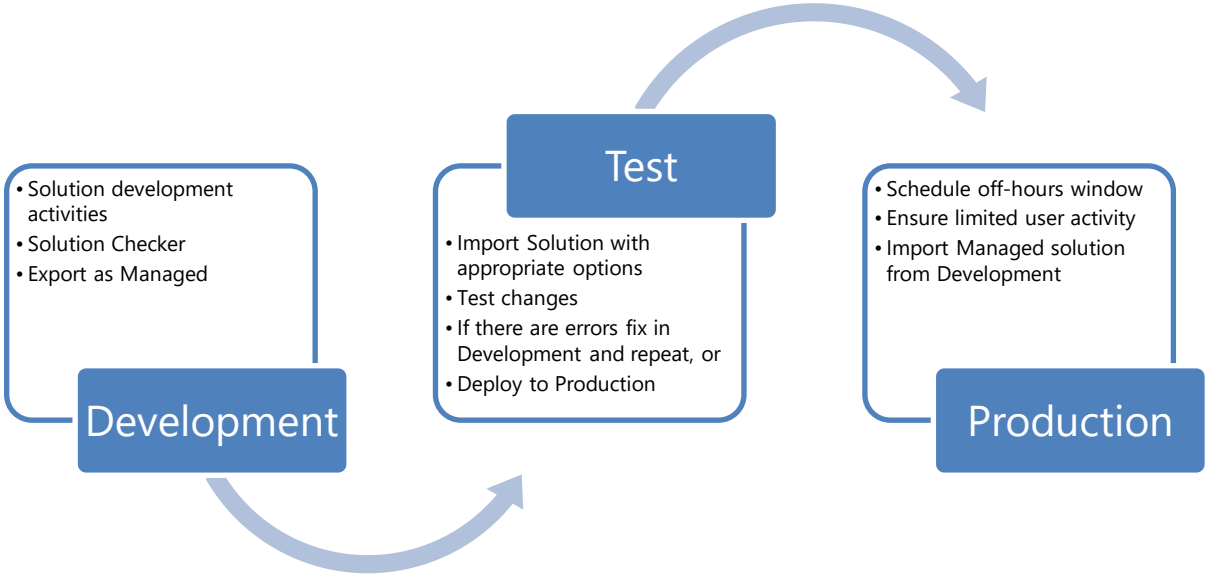
Solution Management Tools – More Info

- 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

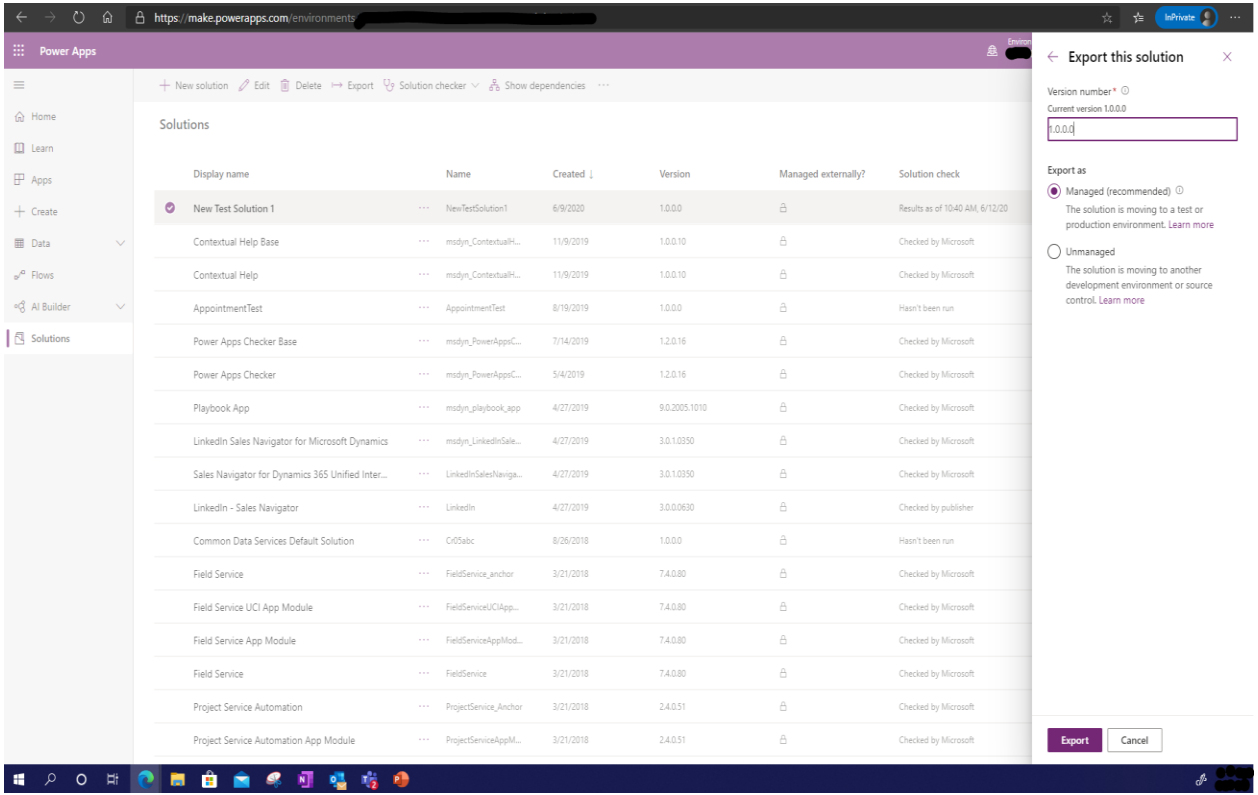
The “Manual” Method

Deploying Manually - Overview



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

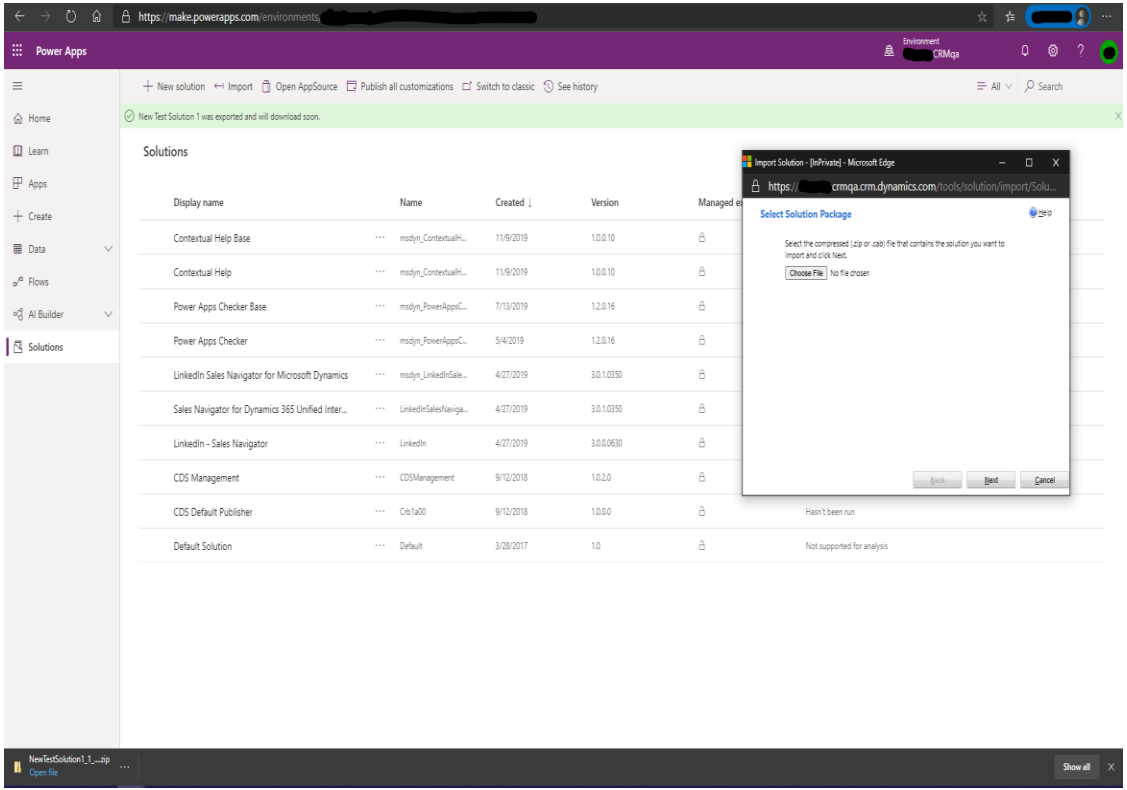
Deploying Manually – Exporting Your Solution



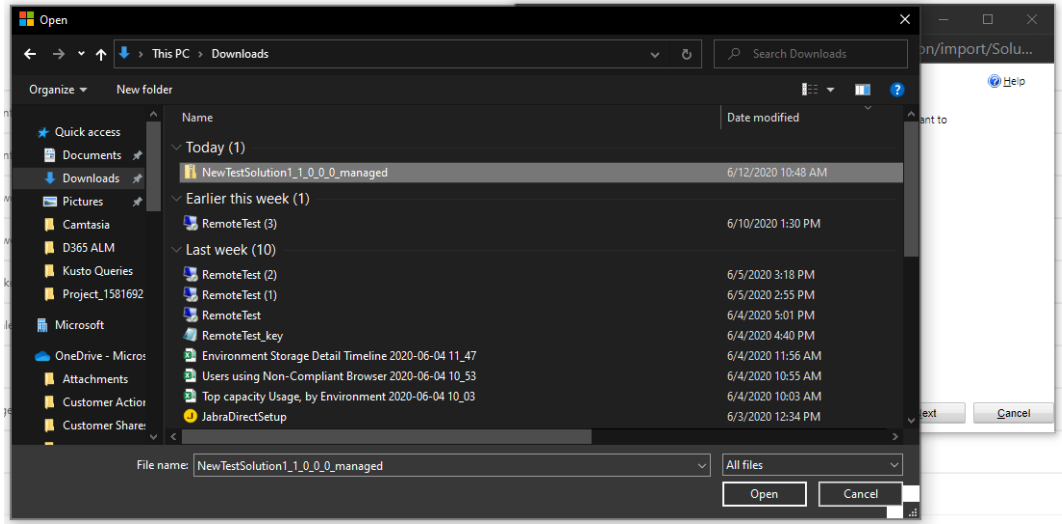
- First step after completing a Solution build in DEV and you're ready for deeper testing (QA/UAT)
- Use Power Apps (make.powerapps.com) to manage your solutions
- Select the solution you wish to migrate and choose Export in the toolbar
- Creates a .zip file that contains all of the Solution XML files
- 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

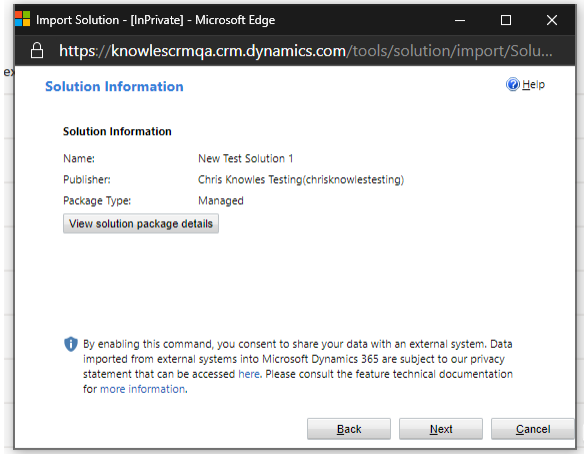


Deploying Manually – Importing Your Solution



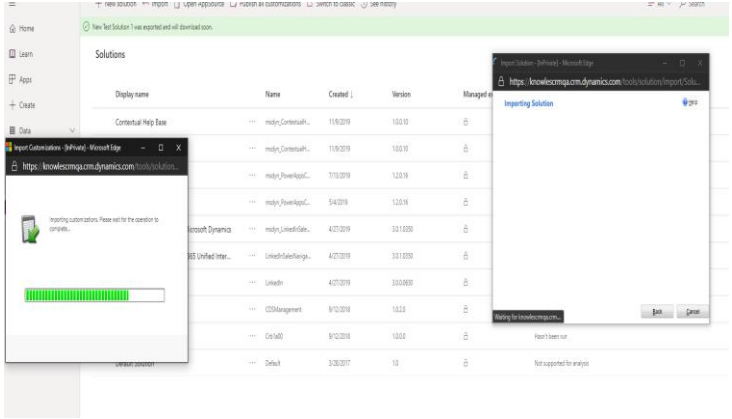
✓ Select your solution .zip file and press the Open button

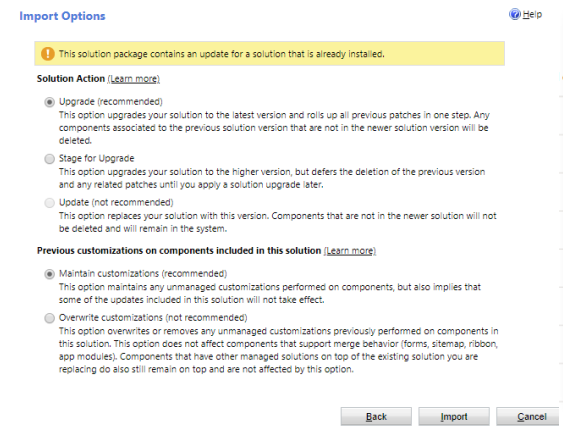
✓ If the correct file was chosen and resembles the screen shown to the right looks, press *Next*

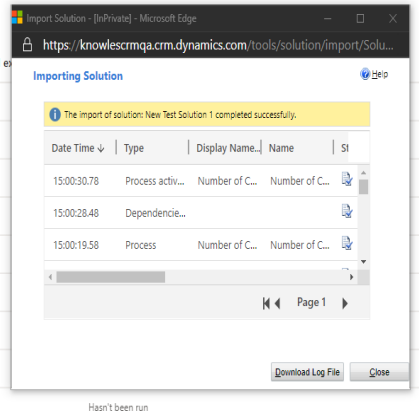


Hasn't been run

Deploying Manually – Importing Your Solution







- 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)

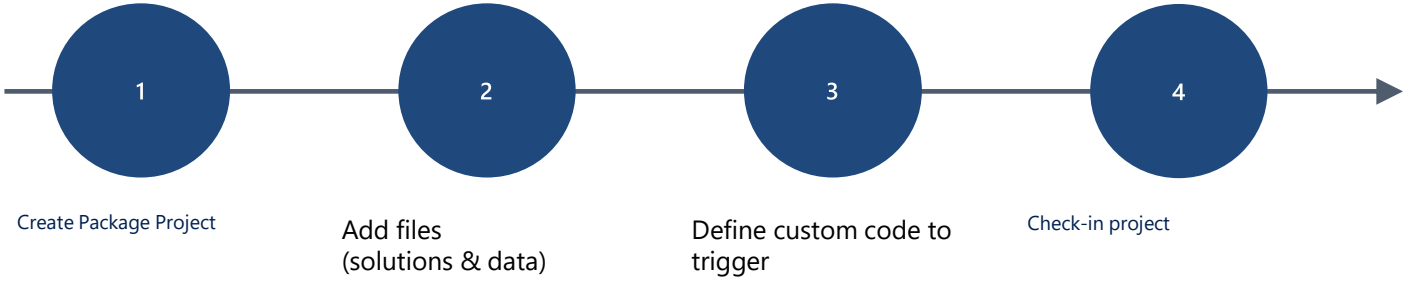
For more information on importing solutions, see https://docs.microsoft.com/en-us/powerapps/maker/common-data-service/import-update-export-solutions

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

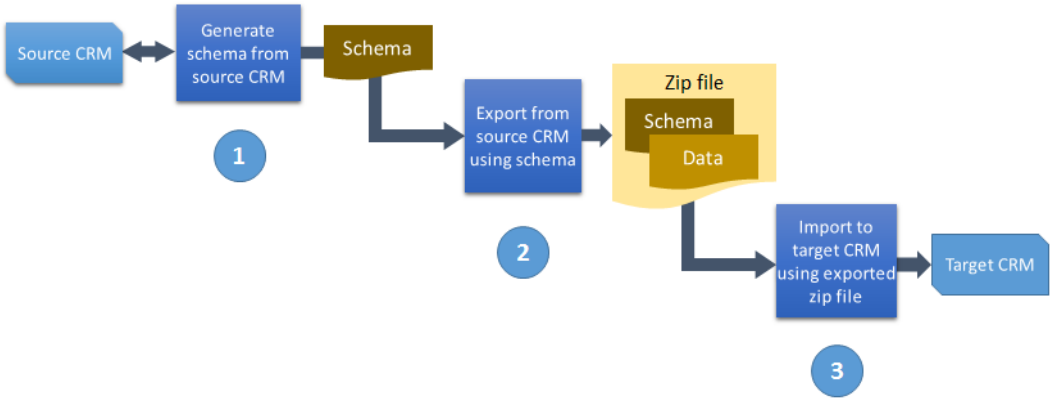
- Find more information on the Package Deployer tool here
- Assists with deploying applications comprising multiple solution and configuration data
- A Visual Studio project template is provided to construct package definitions
- Project can be version controlled and used in build process



Configuration Migration Tool

Configuration Migration Tool

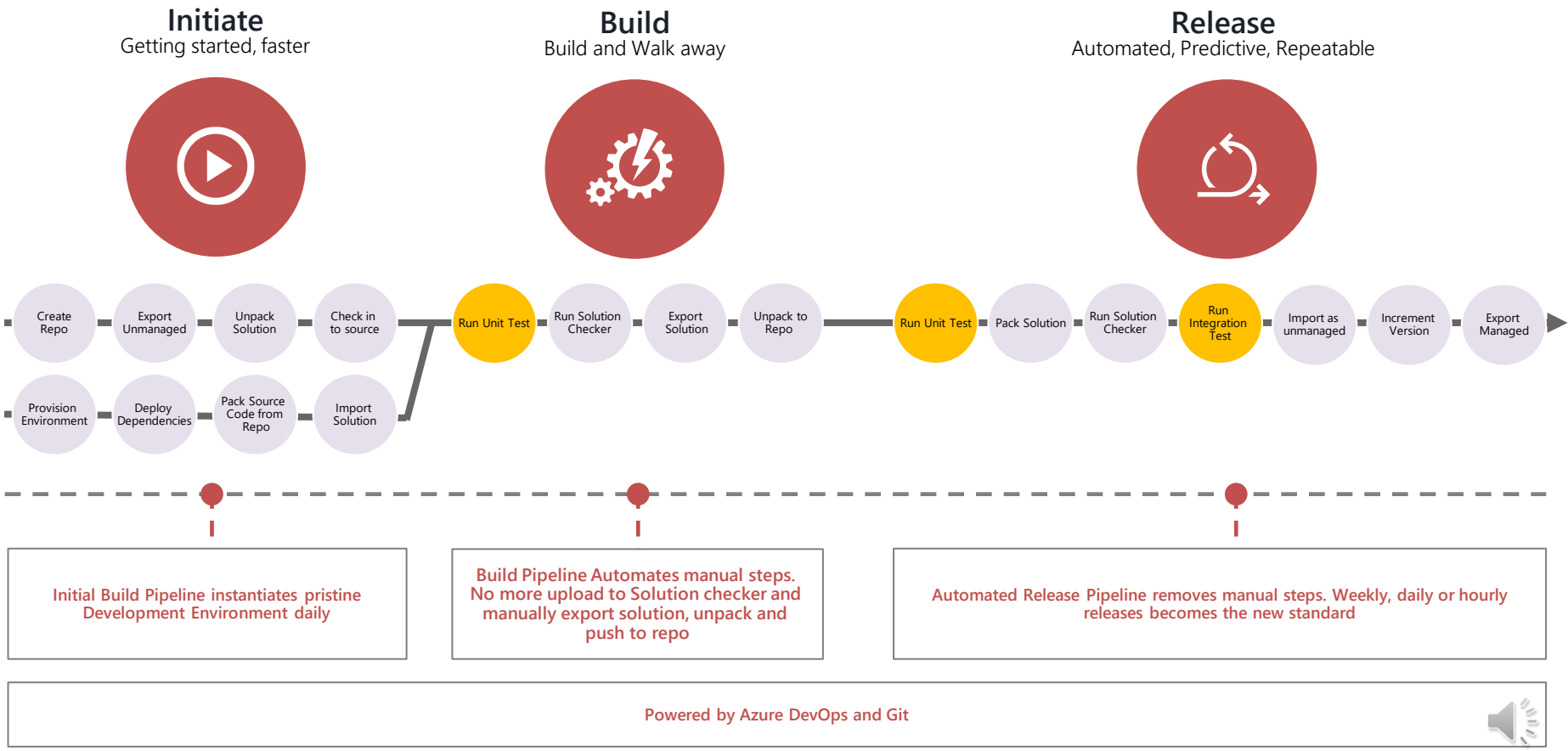
- Designed to migrate custom functionality configuration data from one environment to another.
- The configuration data defined in the Unified Service Desk application for Dynamics 365 is a good example of the entities that would have to be migrated using the Configuration Tool
- Not designed to be a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



For more information, see https://docs.microsoft.com/en-us/power-platform/admin/manage-configuration-data

Tools for Automation

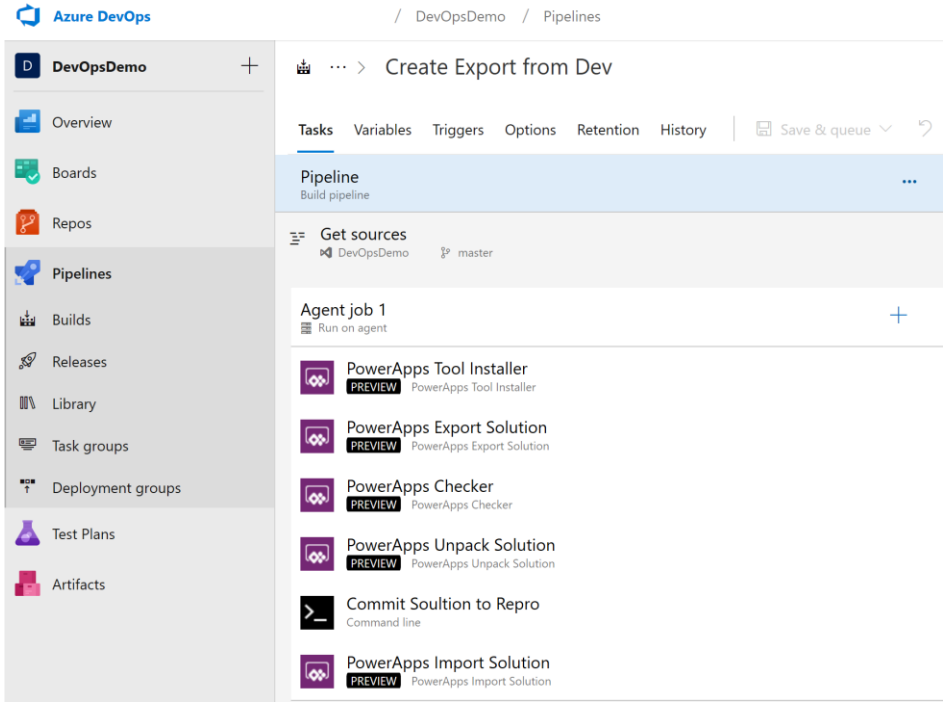
Tools For Automation - Azure DevOps



Tools for Automation – PowerApps Build Tools

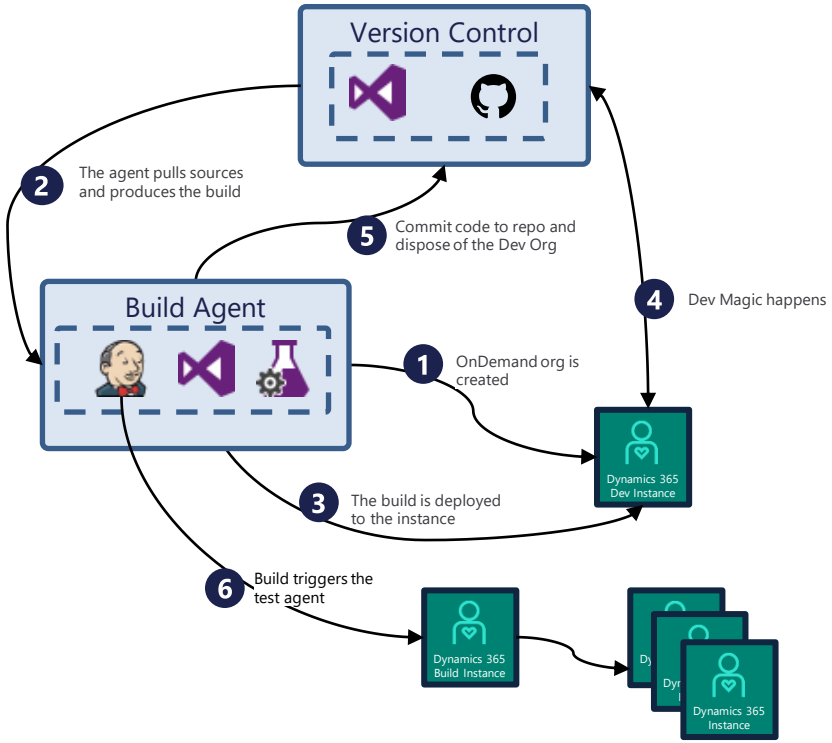
Use PowerApps Build Tools to automate common build and deployment tasks related to PowerApps.

- ✓ 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

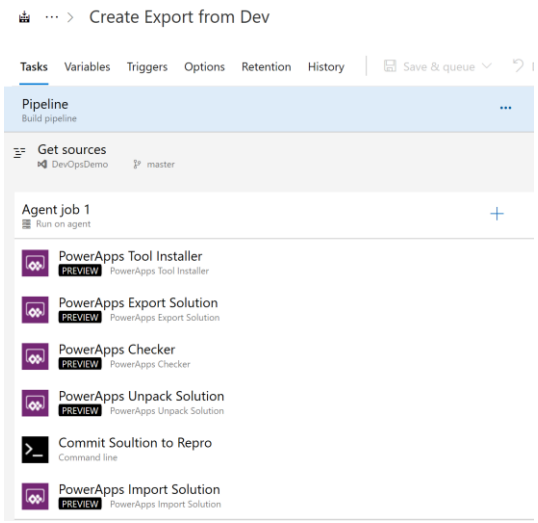


A build is queued and initiates:

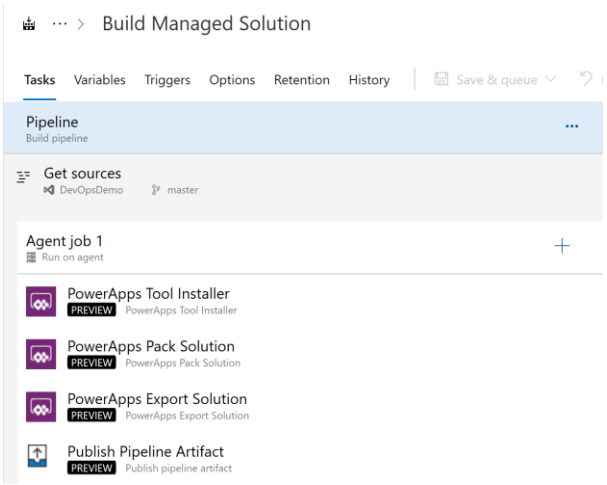
1. Org creation
2. The build agent pulls sources from version control and produces the build
3. The solution or package is deployed to the instance
4. Dev Magic happens
5. Solution checker runs and the code/configuration is pushed to source control and the instance destroyed
6. Run automated tests and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

Build Process

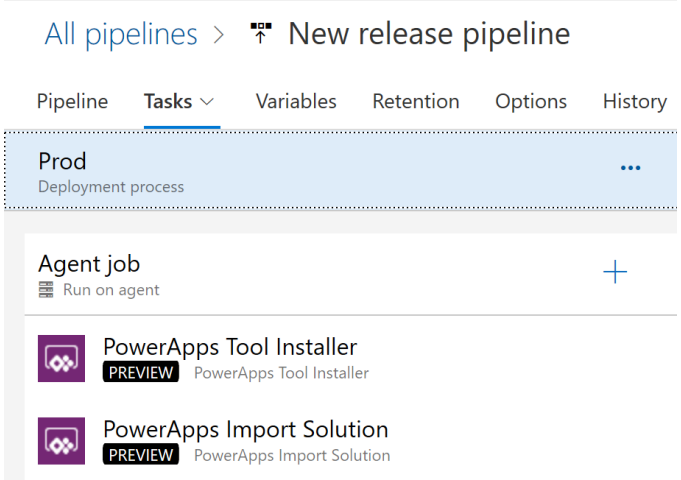
1. Extract from Dev, store in Source Control and import into Build org



2. Extract from build org and create Managed Solution

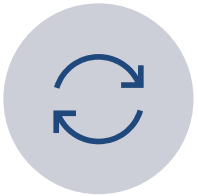


3. Import managed solution in the upstream orgs



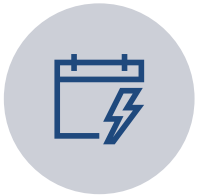
Test Automation with EasyRepro

UI Automation Test Library for MDA - EasyRepro



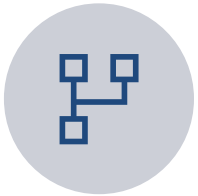
TEST AUTOMATION PROVIDES SIGNIFICANT ROI

INCREASINGLY IMPORTANT FOR “ONE VERSION”



ABSTRACTS TEST CASE IMPLEMENTATION FROM UI CONTROL IMPLEMENTATION

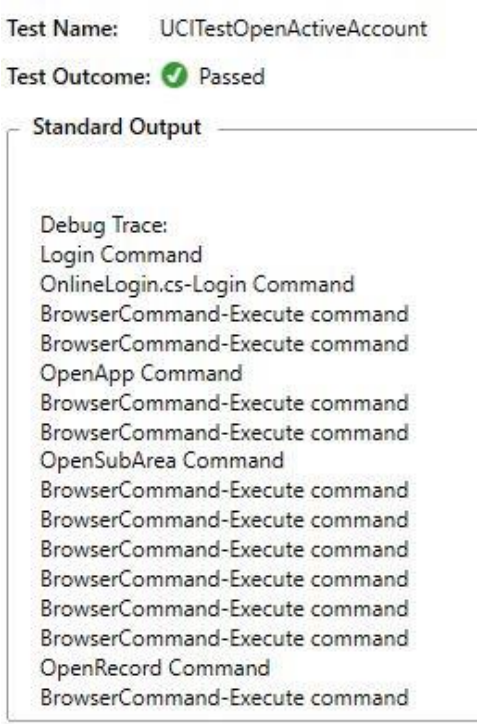
INTUITIVE AND EFFECTIVE TO GET UP AND RUNNING



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!



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:

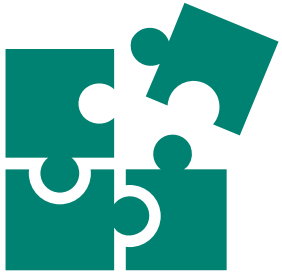
- *Inconsistent development governance*
- *No ability to successfully implement sustainable Source Control*
- *Customer attempts at Automation/CI were complex and ultimately unsuccessful*
- *Significant Customer, Partner and Microsoft effort-hours expended to resolve dependency conflicts or errant code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

Best Practice

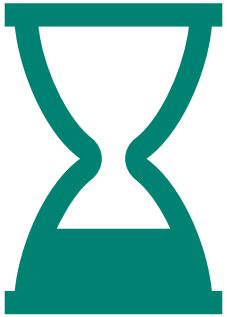
Microsoft recommends
Build Unmanaged, Deploy Managed
as the best practice model for managing Solutions with all Dynamics 365 Legacy and Model-driven Apps.

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?

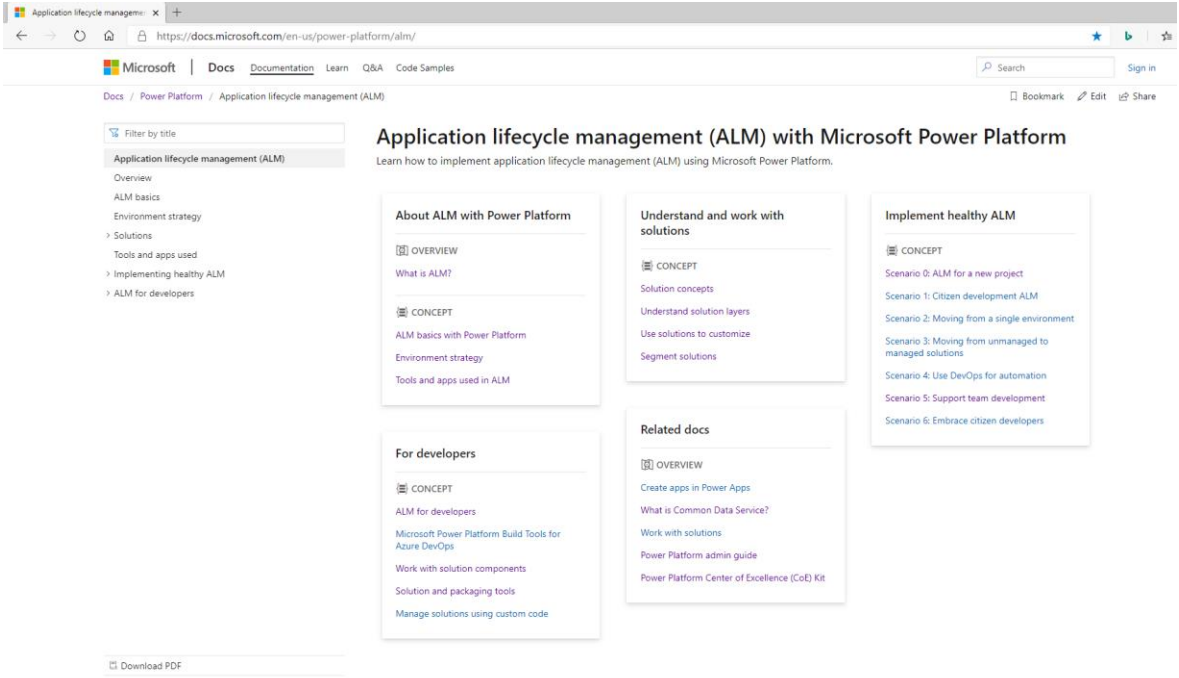
Single publisher?

Solution Checker?

Separate environments for each solution?

Only have your managed dependencies installed?

Next Steps – Read our ALM Documentation



- 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