

ALM: Unmanaged to Managed Solutions Tech Talk

August 2022



Agenda

- Solutions Overview
- Unmanaged vs Managed Solutions
- Get to a Managed Solutions State
- References

Solutions Overview

Solutions Overview

Solutions are used to transport apps and components from one environment to another, or to apply a set of customizations to existing apps.

Solutions include metadata and certain entities with configuration data. Solutions don't contain any business data.

Solutions can contain many different Microsoft Power Platform components, such as model-driven apps, canvas apps, site maps, flows, entities, forms, custom connectors, web resources, option sets, charts, and fields

Solutions are packaged as a unit to be exported and imported to other environments or deconstructed and checked into source control as source code for assets. Solutions are also used to apply changes to existing solutions.

[Application lifecycle management \(ALM\) basics with Microsoft Power Platform - Power Platform | Microsoft Docs](#)

Solution Publishers

Publishers "own"
components

1 Publisher is best
practice

**Can't move
components between
publishers**

Determines the **prefix**
of newly created
components

Solutions Lifecycle

Solutions support the following actions that help support application lifecycle processes

Create Author and export unmanaged solutions.



Update Create updates to a managed solution that are deployed to the parent managed solution. You can't delete components with an update.

Upgrade Import the solution as an upgrade to an existing managed solution, which removes unused components and implements upgrade logic. Solution upgrades will delete components that existed but are no longer included in the upgraded version.

Patch No Longer recommended



Update - Recommended BAU deployment method

[Solution lifecycle - Power Platform | Microsoft Docs](#)

Solution Segmentation Strategies

Solution Segmentation strategies typically involve the separation of components either via Component Type or Business Area, although often there is a combination on the two.

Implementations have often defined segmentation strategies as a workaround to achieve faster solution deployment times to minimize business impact during deployments. Recent investments and subsequent improvements have been made to the solution import performance, reducing the needs of segmentation for performance benefits.

It is recommended to use segmentation for different Apps and/or Business Areas, if they require different development teams and deployment cadence.

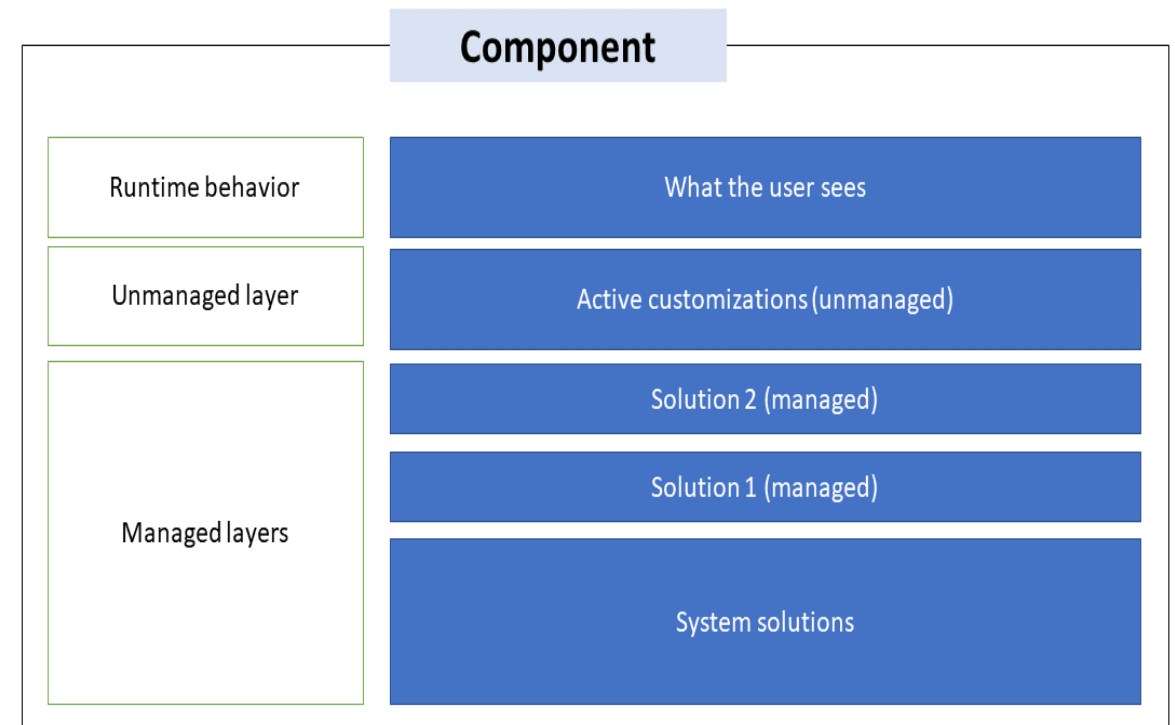
An example of segmentation is how Microsoft packages 1st party solutions, having solutions separated by functional area i.e., Sales, Service, Marketing, etc

Note: some components such as Connection References and PCF controls make sense to be in their own solutions.

Solution Layers & Dependencies

Solution layering is implemented at a component level. Managed and unmanaged solutions exist at different layers within a Microsoft Dataverse environment. In Dataverse, there are two distinct layers, Managed and Unmanaged

- ✓ Layers describe the dependency chain of a component from the root solution introducing it, through each solution that extends or changes the component's behaviour
- ✓ 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.



Unmanaged vs Managed Solutions

Unmanaged vs Managed

A solution is either managed or unmanaged.

- **Unmanaged solutions** are used in development environments while you make changes to your application.
- Unmanaged solutions can be exported either as unmanaged or managed.
- Exported unmanaged versions of your solutions should be checked into your source control system.
- Unmanaged solutions should be considered your source for Microsoft Power Platform assets. When an unmanaged solution is deleted, only the solution container is deleted.
 - All the unmanaged customizations remain in effect and belong to the default solution.

[Solution concepts - Power Platform | Microsoft Docs](#)

- **Managed solutions** are used to deploy to any environment that isn't a development environment for that solution. This includes test, UAT, SIT, and production environments.
- Managed solutions can be serviced independently from other managed solutions in an environment. As an ALM best practice, managed solutions should be generated by exporting an unmanaged solution as managed and considered a build artifact.
- You can't edit components directly within a managed solution. Editing of managed components can only be done in the associated unmanaged solution in dev or by adding the component to an additional unmanaged solution in dev which then will create a dependency between the solutions.
- You can't export a managed solution.
- When a managed solution is deleted (uninstalled), all the customizations and extensions included with it are removed.

Managed Solution Benefits

- ✓ Ability to Uninstall
- ✓ Environment Integrity
- ✓ Dependency Management
- ✓ Solution Layering
- ✓ Solution Integrity (reliable source control)
- ✓ Clearly defined component ownership
- ✓ Improved solution import performance with reduced performance impact on the production environment
- ✓ Better process for eliminating components that are no longer needed
- ✓ **Safer, Repeatable and Reliable Deployments and Testing**

Getting to a Managed State

Prerequisites

- At least two copies of Production (for Dev and Test isolation)
 - Optional Data Hygiene Step: Remove tables and components that are not needed from your production environment. For example, consider deleting tables with no records or no usage
- A new Solution to contain all the unmanaged components
 - Additional solutions can be used for other components such as Canvas Apps, Plugins, PCF, Flows
- Testing strategy,
 - Test Automation?
 - Validation of Managed transition
- Cut over strategy
 - When will in flight development need to be incorporated into the Managed transition
 - Will development processes change
 - Will the project become source control centric with automated builds and releases

Adding components to the newly created solution

- For unmanaged components, such as custom unmanaged tables, you won't use segmentation but will select Include all components when adding those components to the solution.
- If there are managed components that you've customized, use segmentation when adding those components to the solution. For example, if you've added a custom column or changed the display name for an existing column to a Power Apps standard table, such as Account and Contact, use segmentation so you only export the customized components your project needs and not additional components that you don't intend to service. To do this, choose Select components, and then add only your customized components to the solution. **Note** – if you add all assets, you will be continuously be receiving new dependencies introduced from 1st party solutions
- If you have canvas apps, flows, portals apps, or plug-ins to convert, you can add them to a separate unmanaged solution now, or at a later time.
- Use a single publisher for all your solutions.
- XrmToolBox component mover has been used by other customers that have transitioned to managed solutions

[Move from unmanaged to managed solutions to support healthy ALM with Power Apps - Power Platform | Microsoft Docs](#)

Deploy and Test the new solution as Managed

- A. Export the unmanaged solution(s) from your development environment as managed.
- B. If there's an unmanaged solution in your test environment that has the same name as the managed solution you want to import, delete the unmanaged solution record in the test environment. Note: deleting the unmanaged solution doesn't delete the solution's unmanaged components.
- C. Import the solution into your test environment using [Power Platform CLI](#) or [Power Platform Build Tools](#) **solution import** commanding with the **convert-to-managed** parameter or the DevOps tooling solution import task. Currently, you can't successfully import the solution and convert all components to managed using the Maker Portal.
- D. There are various types of tests you can use to check whether specific functions or features of your app are working correctly. Some of these tests include unit tests, end-to-end, and user acceptance tests (UAT).
- E. After testing is completed and validated you can deploy the new solution to Production. If you have used a separate Dev environment for the creation of the new unmanaged solution you will need to deploy this back to BAU Dev environments.
- F. Consider how normal BAU development will be done post the move to managed solutions, generally this transition is part of a wider healthy ALM piece of work that would enable a source control centric approach with automated builds and releases. It is the enabler the safer, faster and reliable deployments.

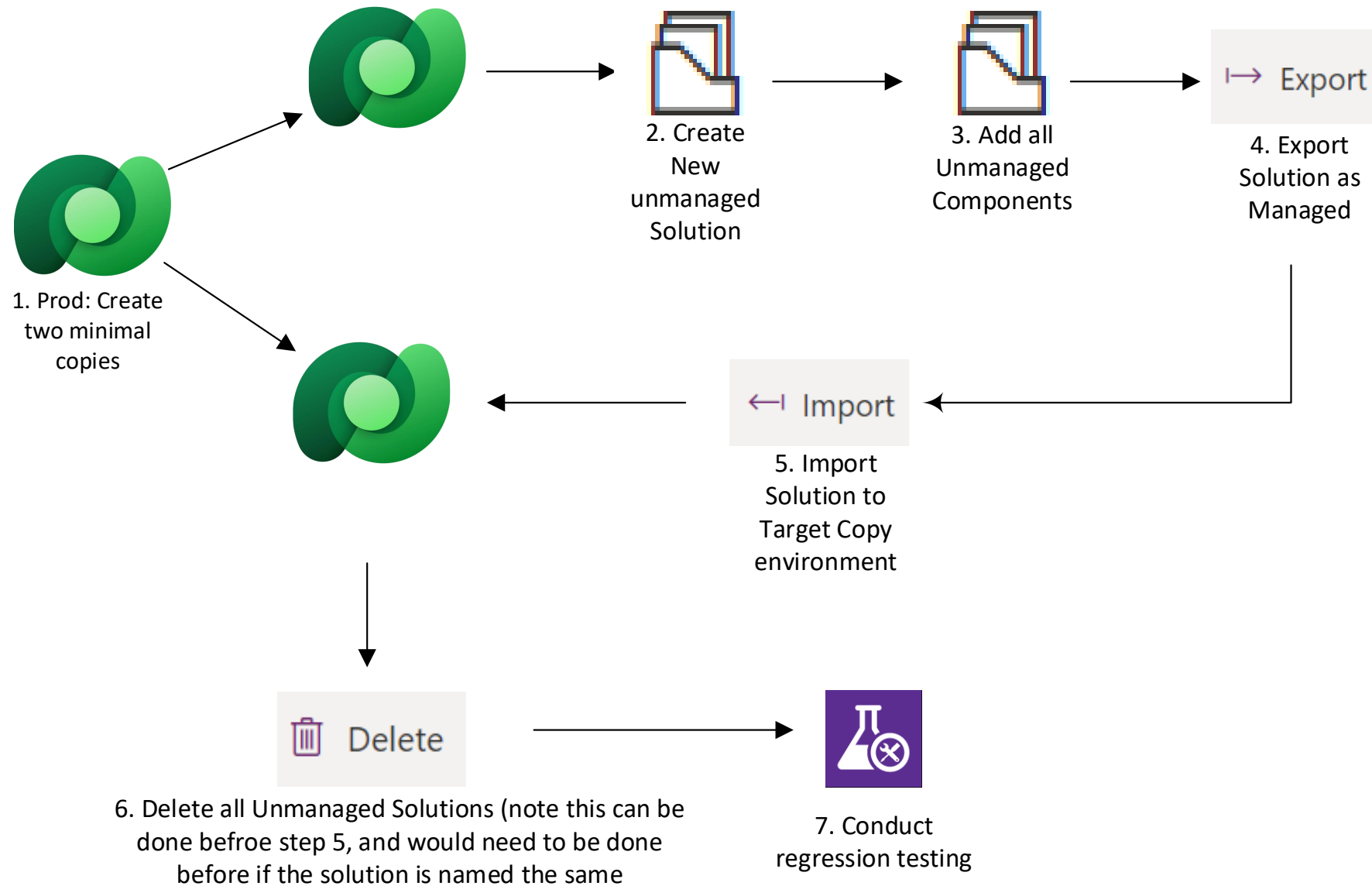
Recap

Identifying the unmanaged changes is the challenge. It is unlikely every unmanaged change will be identified but the goal here is to get customers to a managed state and then always deploying managed solutions.

Process Overview

1. Take **two minimal copies of production**, one for creating the new unmanaged solution(s) and one to deploy the new solution as managed and conduct testing
2. Create new Unmanaged solution(s) and add the required components, ensure the solution(s) has the same custom publisher
3. Delete the unmanaged solutions from the target (this can also be done after import)
4. Export the new unmanaged solution(s) as managed and import to the target environment via a pipeline using the **convert to managed** option
5. Conduct full regression testing
6. **Repeat the process** until you can validate all unmanaged components have been converted to managed
7. Import Managed Solutions to all non dev environments (Test, Prod etc) and delete unmanaged solutions*

*you cannot deploy a managed solution into an environment with the same unmanaged solution name



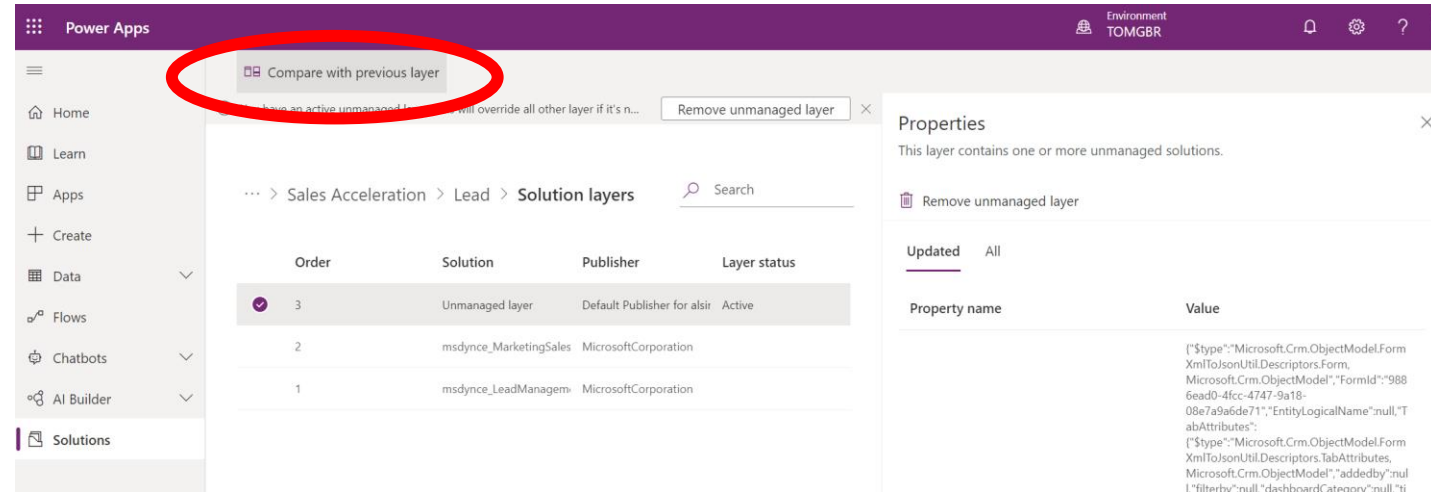
Identifying unmanged components

`/api/data/v9.2/msdyn_solutioncomponentsummaries?$select=msdyn_ismanaged,msdyn_solutionid,msdyn_name,msdyn_displayname,msdyn_componenttypename,msdyn_schemaname,msdyn_objectid,msdyn_componentlogicalname,msdyn_primaryidattribute&$filter=msdyn_ismanaged eq 'false' and msdyn_statusname eq 'Active' and (msdyn_componenttypename ne 'Environment Variable Value' and msdyn_componenttypename ne 'Connection Reference')`

- Can be used to identify components in an unmanaged state (this is not a catch all)
- Can be used within a browser when logged into Dynamics or alternatively via Postman

Review Solution Layers

- Review Solution Layers contributing to a component's customizations
- Review the XML for the specific layer's customization
- Compare the XML difference from previous Layer
 - Works on Layers with an order $\neq 1$
 - Not available for all components



Power Apps Environment TOMGBR

Compare with previous layer

Remove unmanaged layer

Sales Acceleration > Lead > Solution layers

Order	Solution	Publisher	Layer status
3	Unmanaged layer	Default Publisher for alsir	Active
2	msdynce_MarketingSales	Microsoft Corporation	
1	msdynce_LeadManagem	Microsoft Corporation	

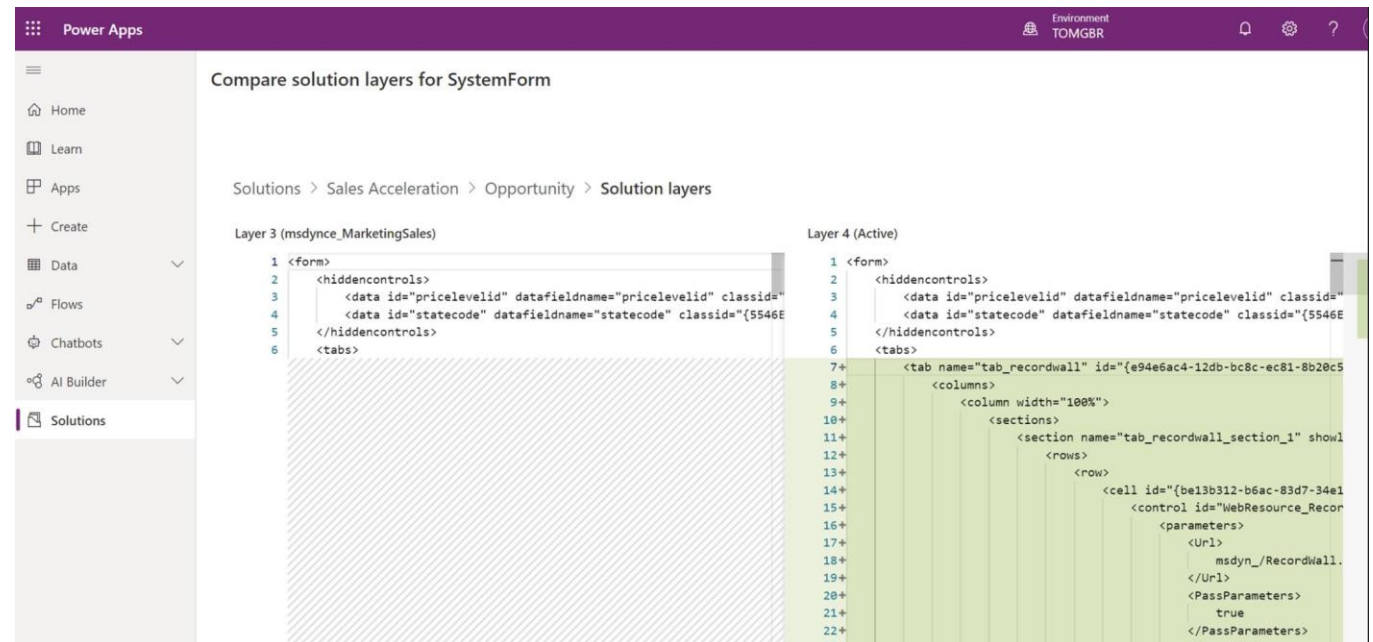
Properties

This layer contains one or more unmanaged solutions.

Remove unmanaged layer

Updated All

Property name	Value
	["\$Type": "Microsoft.Crm.ObjectModel.Form XmlToJsonUtil.Descriptors.Form, Microsoft.Crm.ObjectModel", "FormId": "9886ead0-4fcc-4747-9a18-08e7a9a6de71", "EntityLogicalName": null, "TabAttributes": ["\$Type": "Microsoft.Crm.ObjectModel.Form XmlToJsonUtil.Descriptors.TabAttributes, Microsoft.Crm.ObjectModel", "addedby": null, "filterbv": null, "dashboardCategory": null, "ti



Power Apps Environment TOMGBR

Compare solution layers for SystemForm

Solutions > Sales Acceleration > Opportunity > Solution layers

Layer 3 (msdynce_MarketingSales)

```
1 <form>
2   <hiddencontrols>
3     <data id="pricelevelid" datafieldname="pricelevelid" classid="
4     <data id="statecode" datafieldname="statecode" classid="{5546E
5   </hiddencontrols>
6   <tabs>
```

Layer 4 (Active)

```
1 <form>
2   <hiddencontrols>
3     <data id="pricelevelid" datafieldname="pricelevelid" classid="
4     <data id="statecode" datafieldname="statecode" classid="{5546E
5   </hiddencontrols>
6   <tabs>
7     <tab name="tab_recordwall" id="{e94e6ac4-12db-bc8c-ec81-8b20c5
8     <columns>
9       <column width="100%">
10        <sections>
11          <section name="tab_recordwall_section_1" show1
12          <rows>
13            <row>
14              <cell id="{be13b312-b6ac-83d7-34e1
15              <control id="WebResource_Recor
16                <parameters>
17                  <Url>
18                    msdyn_/RecordWall.
19                  </Url>
20                <PassParameters>
21                  true
22                </PassParameters>
```

References & Documentation

References & Documentation

General ALM Documentation

[Application lifecycle management \(ALM\) with Microsoft Power Platform - Power Platform | Microsoft Docs](#)

[Use scenarios to implement healthy ALM with Microsoft Power Apps - Power Platform | Microsoft Docs](#)

[Application Lifecycle Management | July 13, 2020 - Microsoft Dynamics Blog](#)

Unmanaged to Managed

[Move from unmanaged to managed solutions to support healthy ALM with Power Apps - Power Platform | Microsoft Docs](#)

[Migrating To Managed Solutions At Microsoft - Power CAT Live – YouTube](#)

Success Story

[How a Microsoft team transitioned from unmanaged to managed Power Platform solutions | Microsoft Power Apps](#)

Thank You