Power Platform
Dynamics 365 Model Driven Apps

Customization and Configuration Guidelines

Document Information

Authors	Michael Dietzsch, Armin Fischer, Jens Grambau, Alexander Soutchilin, Markus Konrad, Anne Stanton, Dave Corun
Status	Release

History

Version	Date	Modified by	Changes
17	04.11.2020	Anne Stanton	• Updated to reflect Power Platform vocabulary • Updated grammar and sentence structure • Added Accenture/Avanade Notes indicator in the appendix and updated all "notes" in the document • Reviewed through page 42 out of 50. • Recommend a screen shot refresh and incorporating links to Microsoft Docs where applicable now that Microsoft Docs are mostly updated to 2020. • Microsoft has deprecated the term CE, removed CE.
18	05.11.2020	Anne Stanton	• Reconciled documents
19	02.02.2021	Markus Konrad	• Added details for Business Units (26)
20	16.02.2021	Michael Dietzsch	• Added Translation details
21	10.03.2021	Michael Dietzsch	• Renamed tables, columns etc. and changed overall texts everywhere
22	11.03.2021	Michael Dietzsch	• Added classic editor, changed texts
23	08.12.2021	Michael Dietzsch	• Added naming conventions for column names, plus renaming paragraph
24	09.12.2021	Armin Fischer	• Added handling instructions for Publishers • Updated info on solution structure and slicing • Added general alignments to Avanade BizApps Core
25	14.12.2021	Dave Corun	• Minor updates around Organization owned entities, field length changes, and UCI form limitations
26	10.02.2022	Armin Fischer	• (Re-added) guidance to always add a description to Columns (10.2).

2	05.05.2022	Michael Dietzsch	• Corrected casing for schema names of Two Options columns.

Table of Content

1 Description

This document provides guidelines for customizing and extending Microsoft Dynamics 365 Model Driven Apps on the Power Platform. It should be used to support project consistency and to align project team members who will be customizing and extending the system applications.

By adhering to the guidelines, you are helping your coworkers to create consistent projects and reducing project risks.

These guidelines are an integral part of the Avanade Delivery Framework (ADF) for Business Applications .

The guidelines for customizing mentioned in this document apply to both D365 Online and D365 On-Premise versions. Screenshots are taken from the Online version, the On-premise version will look different, but the functionalities are the same.
Some specific capabilities maybe not available On-Premise if major discrepancies are known they will be stated in this document.

2 Terminology Updates

Microsoft changed some terminologies. These changes are available here.
Please be aware that these changes are not consistently updated in all places.

Legacy term	Current term
Entity, entities	Table, tables
Field, fields Attribute, attributes	Column, columns
Record, records	Row, rows
Option set, multi select option sets Picklist, picklists	Choice, choices
Two Options	Yes/No

Note
Terminology changes do not apply to APIs or messages. CreateEntityRequest will not change.

3 Access Customizations

To start customizations, you should be aware that there are currently two options.

The first is the Power Platform Maker Portal which can be reached using https://make.powerapps.com. The maker portal is the newest portal for configuring the Power Platform including all model driven apps such as the Dynamics 365 apps. It has additional features that help increase consistency and should be used for all work unless the work cannot be delivered with the functionality available.

The first step after entering the maker portal is to carefully align to the environment you want to be working in. You can see your available environments in the Upper Right-Hand corner. (1).

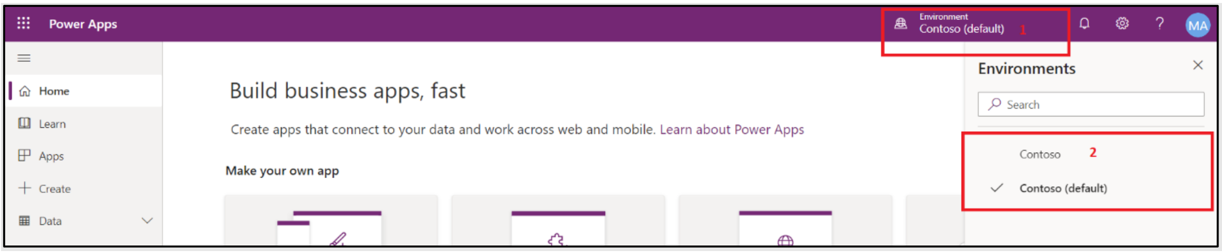


Figure 1 Environment selector

You can change the environment (2) by selecting the environment name.

Note
Not every configuration is available in the new maker portal, therefore; you must use the classic editor from time to time. Over the next few months, we anticipate that this will slowly change. In most cases the classic editor is available as either a link or you can switch directly to it from the maker portal.

Examples where the classic editor is required:

- Changing icons on tables

3.1 Classic Editor and access customizations On-Premise

For On-Premise customizations are currently still done completely in the classic editor. On legacy UI you can access the customization via “Settings” >> “Solutions”.

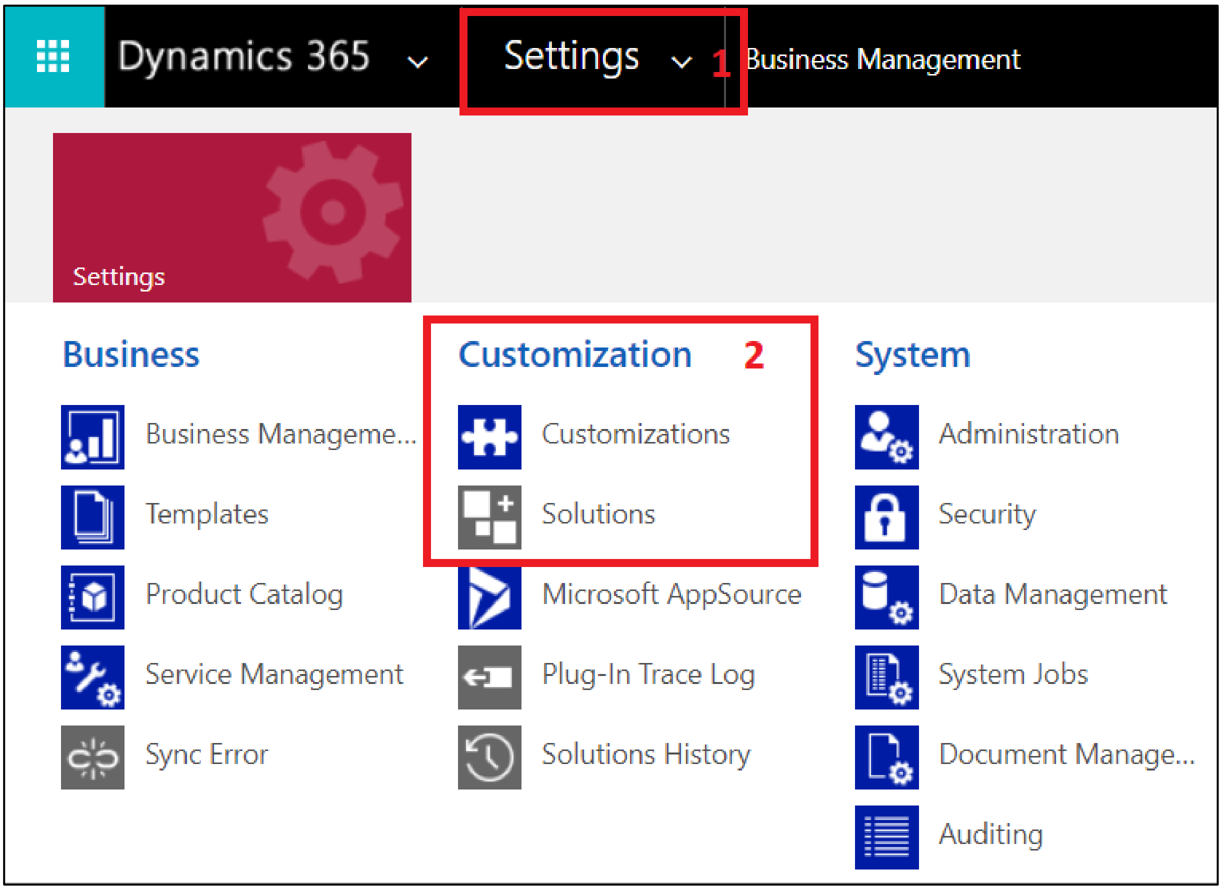


Figure 2 Access Classic Editor On-Premise

Or if you are using UCI UI via the cogwheel. “Advanced Settings” >> “Customizations”

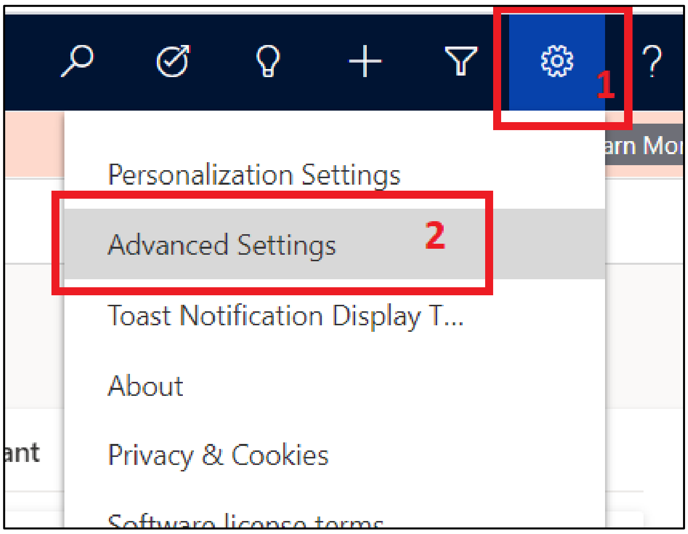


Figure 3 Access customization on UCI

After clicking on solutions, you see the list of currently available solutions.

avanade

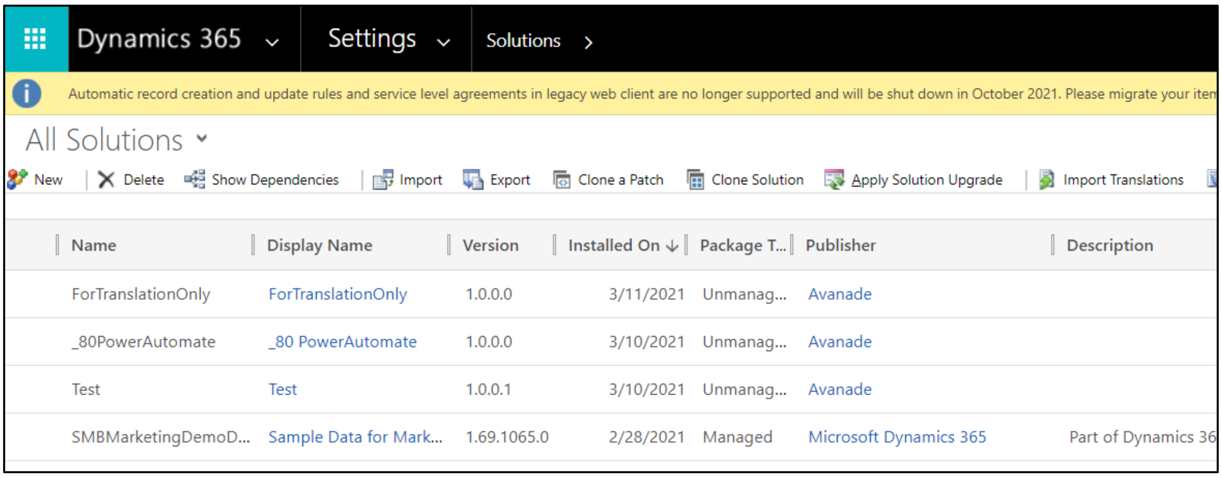


Figure 4 Classic Editor list of solutions

If you click on any solution the classic editor opens.

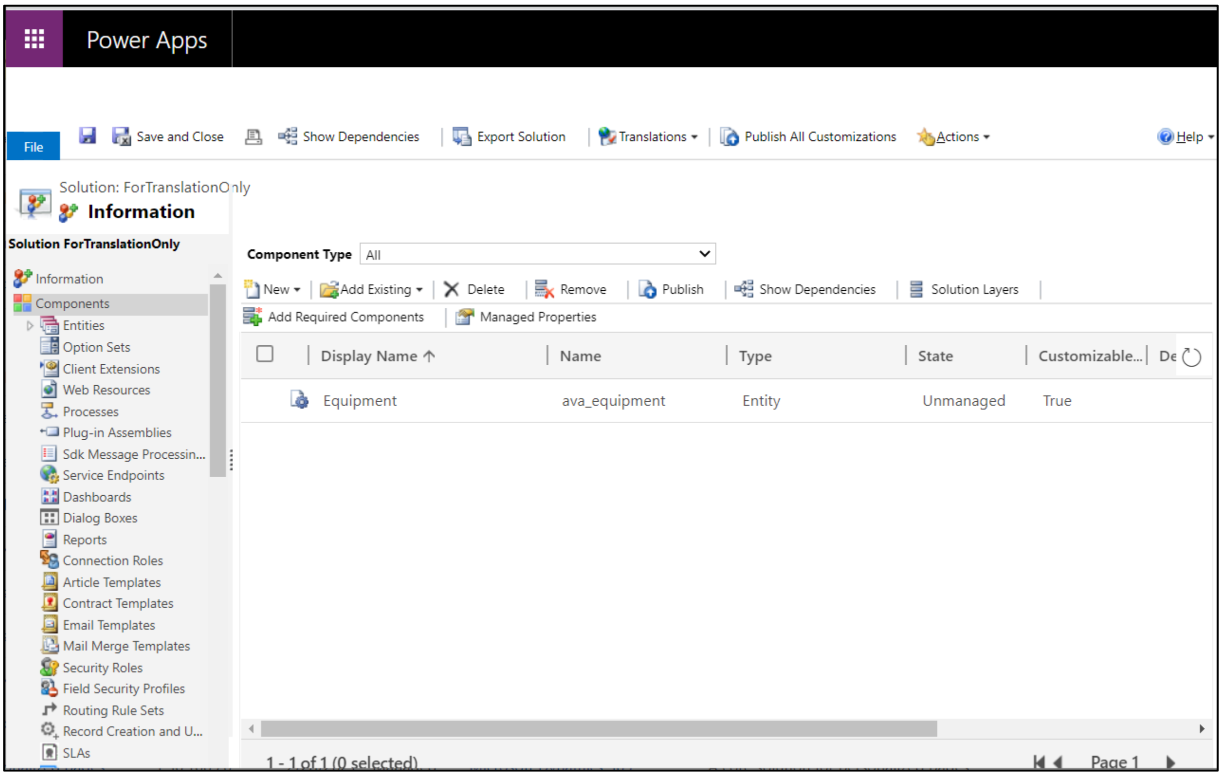


Figure 5 Classic solution editor

4 Create the Publisher

A Publisher is set inside a Solution and among other properties, it defines the customization prefix used for example to set the logical name of a Table. The definition of a Publisher is slightly different in a project context than in an asset context and requires several considerations, especially not to run into issues with managed solutions. Please see chapter 23 for more information.

5 Fits and Gaps of Dynamics 365 Online versus On-Premise

Any Software as a Service offering “Online” has potential restrictions which help different customers and different project teams to be good neighbors in a shared environment. Therefore, Microsoft defines limits on items such as API calls, so that the resources are evenly distributed.

Please reference the Microsoft Documents library for the most up to date information on any online restrictions. A good starting point are the official documentation.
PowerApps documentation: https://docs.microsoft.com/en-us/powerapps/
Power Platform documentation: https://docs.microsoft.com/en-us/power-platform/
Dynamics 365 documentation: https://docs.microsoft.com/en-us/dynamics365/
Dynamics 365 CE On-Premise documentation: https://docs.microsoft.com/en-us/dynamics365/customerengagement/on-premises/overview

There is no official Microsoft page available where the restrictions of Online vs. On-premise are documented. From time to time Microsoft has a page available for specific apps but these pages are not updated regularly and therefore not part of this document.

Additionally, the Microsoft SaaS offering for Dynamics 365 includes benefits and fits that are not available in the Microsoft Dynamics On-premise offering. These align to all features that are leveraging the power of Azure for their functionality such as Customer Insights or Customer Service Insights. It also includes items such as Relevant Search which uses the more powerful Azure search engines and options.

The guidelines for customizing mentioned in this document apply to both D365 Online and D365 On-Premise versions. Screenshots are taken from the Online version, the On-premise version will look different, but the functionality is the same.
Some specific capabilities maybe not available On-Premise if mayor discrepancies are known they will be stated in this document.

6 Protected Words

The Power Platform is a developed solution, and the product team uses certain vocabulary and terms for their system tables. As a project team we need to be aware of vocabulary which is protected and should not be used ("protected Words") otherwise code generation will potentially break.
Therefore, do not use the following words as table or column names:

- System
- Object
- Class
- Abstract
- Interface
- Namespace
- Base
- String
- Integer
- Double
- Float
- Type
- Microsoft
- Feedback
- Activity
- Or any of the Out of the Box Table names such as Opportunity, Lead, Account, contact, Incident, Case, etc.

Furthermore, don't use these words alone as a schema name.

Example:

- Bad - Label: System | schema name: ava_system
- Preferred - Label: Customer System | schema name: ava_CustomerSystem

7 Customizing Standard Tables

- If you change standard tables, always copy the standard forms. **Don't customize the out of the box forms.**

Note
This reduces the risk that a weekly update from Microsoft might break your user experience.

- Name your form with a unique name so that it is not confused with the out of the box forms.
- Also remember to configure your apps to use the right forms and if needed apply the correct form order and security role setup.

Recommendation
Add "(Not in use)" to the label of standard forms which are not use. In this way you can distinguish between forms where customizations are being done and where not.

Deactivation of forms could also be used. But this status is not deployed via solutions.

8 Creating a New Table

8.1 Naming

- Use **pascal casing** for table **schema names** (capital letter for the first letter of each word), e.g. prefix_ReturnOrder instead of prefix_returnorder.
- Don’t use the same schema names as out of the box schema names (for example do not create a table with the schema name “account”).
- The **DisplayName** of a custom table **must be unique** in an organization for the main reason not to confuse tables during Data Import, in Advanced Finds, Workflows, and for Excel Export, etc.
- It must be **different** from the standard **DisplayName.**
- The plural and singular form of a table name must be similar. For example:
    - Good:
        - Singular: Opportunity
        - Plural: Opportunities
    - Bad:
        - Singular: Opportunity
        - Plural: Sales Projects
- For additional restrictions please also look in section 10, Creating New

8.2 Properties

- Deactivate ALL unnecessary options (marked with a +): Notes, Connections, etc. You will be able to reactivate them later!

Properties are divided in the following sections:

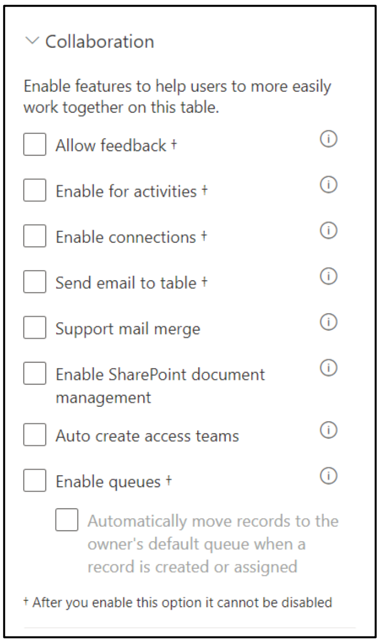


Figure 6 Table options Collaboration that cannot be deactivated

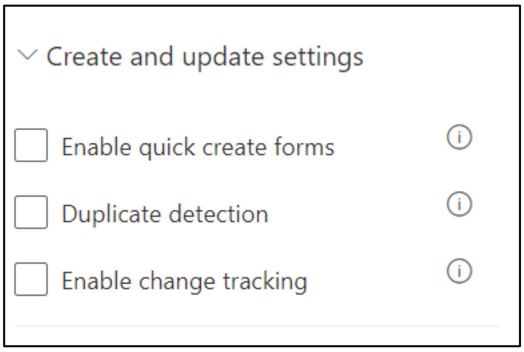


Figure 7 Table options Create and update settings

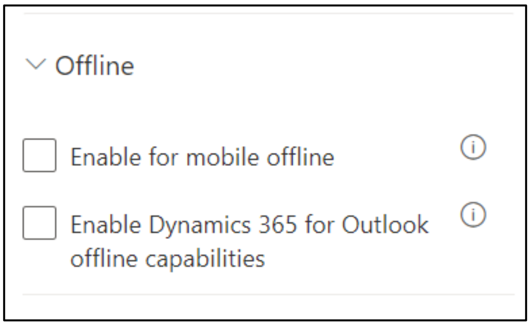


Figure 8 Table options Offline

- Choose the most relevant ownership (see section 8.4 Ownership).

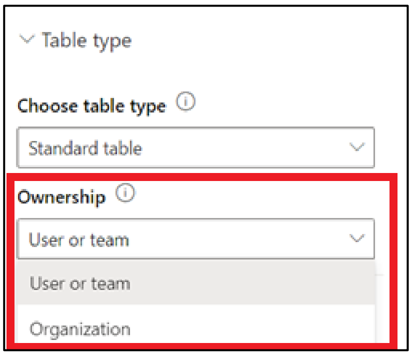


Figure 9 Ownership setup of table

Note: The **activation of any table settings with the cross symbol** + **can't be disabled** afterwards. For this reason, the checkboxes must be unchecked by default and only be enabled if a dedicated user story defines the need for it. The background is that each setting requires resources and has effects in the user navigation which are not always requested. For Example: The activity checkbox does not only create the activity navigation in the left pane. It also makes this table available in all regarding lookup dialogs for all activities.

8.3 Primary Field

8.3.1 Considerations

The Primary Column is prefilled as “Name”.
Once the table primary field is created, the schema name cannot be changed. This also applies to all fields created.

Most clients do not understand the “Name” field until it is presented in a demonstration. Be prepared to discuss how the name field is used as the lookup display.

> ***Important note***
> The column type of the Primary column will be text and is not changeable! The standard maximum length applied will be 100 characters and can be changed up until max 4000.
>
> Since it is of type text you can change it after saving to the type auto number if needed.

The content of the primary name field will be displayed in **the lookup field of the table**.
Do not remove the Name column from a lookup view.
The content of this field will be displayed in **error messages** of this row (e.g. in the **Offline Client** if a synchronization error for this row occurs.

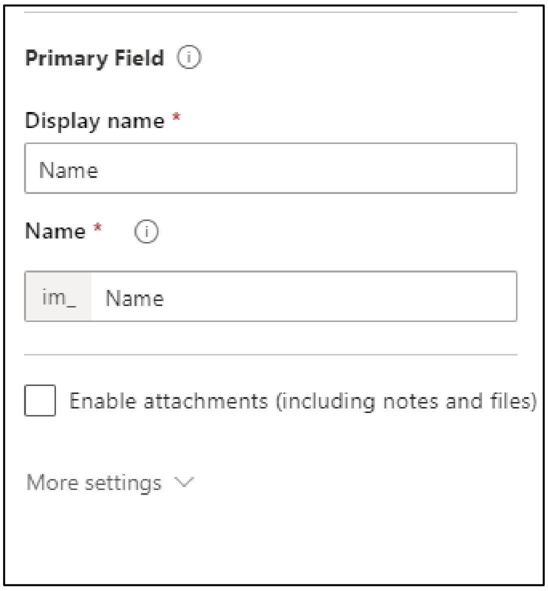


Figure 10 Primary Field

Name this field depending on the business requirement of the table you are creating.

Don’t hide the name on the form if not used!
If the primary column is not used, create an implementation to set this field to a standard! The recommendation is to concatenate one column plus something unique such as the current date
<columnValue> : <Date.Now (yyyy.MM.dd hh:mm)>

8.3.2 Naming

- Name is the default and is what is generally expected.
- If the primary column defines special content: e.g. “ISO3 Code” in country, or “Serial No of the Equipment”, it should be named as “Serial No”, or “ISO 3 Code”. And the schema name should be identical.

avanade

- Exception if the content should have a dedicated data type (e.g. int), then it should be defined in a special data field and synchronized via something like a concatenation plugin.

8.4 Ownership

The ownership defines the security context in which the table will be used. This security context can't be changed later. Therefore, if in doubt use "User or Team". User or Team allows for long term growth with security in mind and the ability to filter data by business unit or user or team.

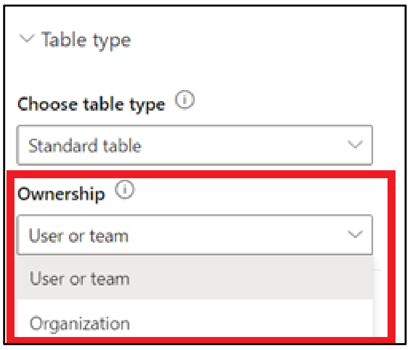
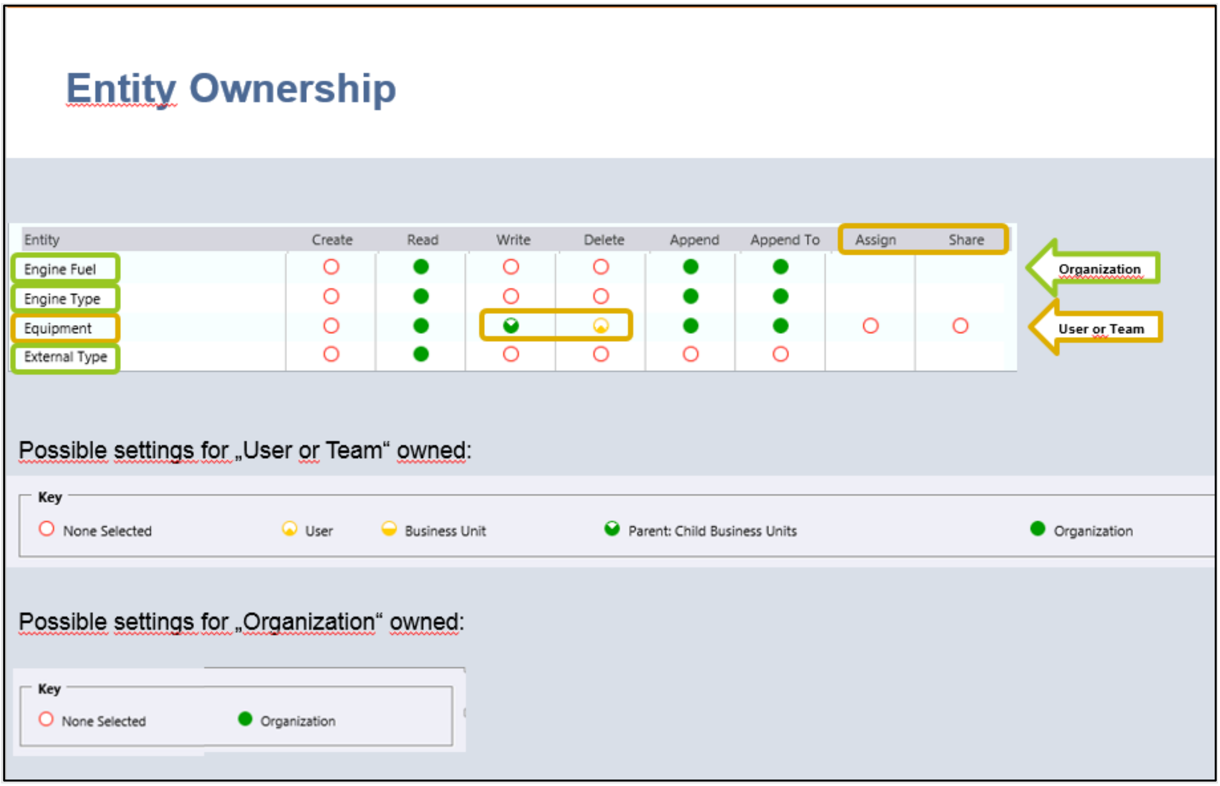


Figure 11 Ownership setup of table

8.4.1 Overview



8.4.2 Organization

Effect:

The table has NO "Owner" field. It has no Assign and Share Operations.

Benefits:

- Easy to configure in the security settings: (No Assign, No Share Setting, "Only None" or "Organization Wide"!

- No Bulk “Reassign Row” is needed if user leave the company. No logic for Team definition is needed

Disadvantages:

- No possibility to set a business unit security level! All rows are readable for every user in the organization if the user has read rights. High cost when this should be changed at a later point in time!
  - The only way to convert an Organization owned entity is to remove it from all relationships, views, dashboards, etc. create a new entity, migrate the data, and then delete the Organization owned entity. Note that if the customer has personal views set up using this entity they may break as well. It is a very high cost to re-create an entity, so it is often safer to create as User or Team owned and this is the default.
- No possibility to “hide” rows of the same table type from different users!
- No ability to filter a subset of lookup data to a specific business unit.
- For example: A business unit might want to only work with a subset of countries.

8.4.3 User or Team

Effect:
The table has an “Owner” field. It has “Assign” and “Share” operations

Benefit:

- Security settings by row owner are possible. Business unit security levels.
- All changes in security requirements can be configured.

8.4.4 Summary

- It is not possible to change a table ownership at some future date. If requirements occur in the future, which need business unit security settings on an organization owned table, then major refactoring will be costly and the requirement will not be met.
- The table would have to be deleted and recreated again. All Fields, View, Forms, JScript’s, Plugins, Reports, Offline Filters etc. must be done again.
- All existing data needs to be recreated.
- **If it is clear case for Organization ➔ Set Organization. If any doubts exist set User or Team!**

8.5 Dependencies

In a usual project, not all table and column definitions in Dynamics 365 are independent and can be considered isolated. Whilst they can be specified and configured relatively quickly, they rather depend on other systems, manual or automated processes or even other Power App solution components (e.g. plug-in code or Azure components). You should think about having a readable dependency list or diagram (can also be automatically created from your repository) which you extend continuously and consult with every customization change to avoid unexpected side effects (for example when you have a reporting system relying on your field definitions).

The Power Platform includes several different Microsoft Development teams who are writing code daily. Please consider these teams your peers as you extend and develop on the platform.

9 Creating Relationships

9.1 Many:1 and 1: Many

Microsoft prepopulates the Lookup column name with the display name.

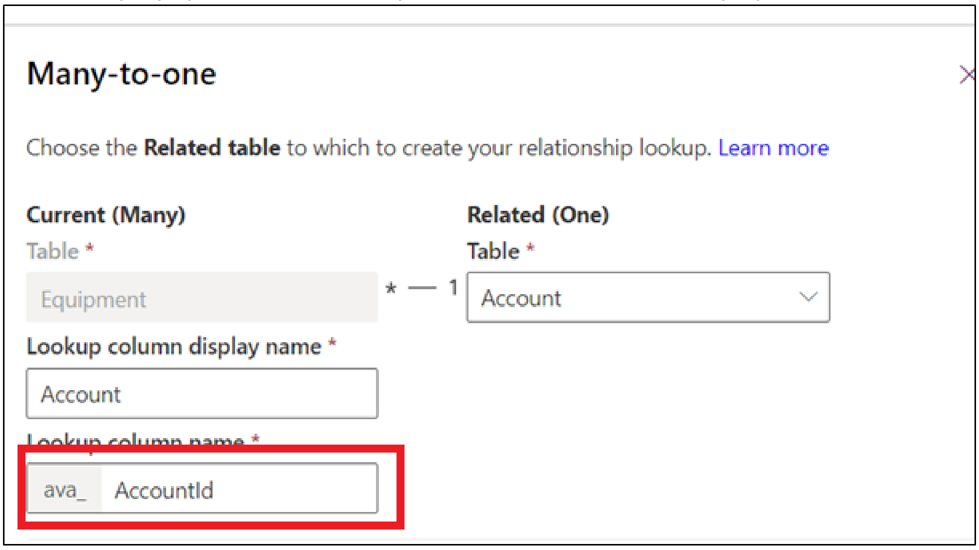


Figure 12 Lookup column name

Important Note

Add “Id” at the end of the Lookup column name. This is currently not done anymore in the new editor. It will help you later when you must create reports or must document tables etc. when you can distinguish between foreign keys and not. Because in these editors you only have the schema names available.

9.2 Naming

MS prepopulates names for relationships e.g.: prefix_lookupschemaname_relatedtableschemaname. Prefixes are removed. So, you get ava_AccountId_Account

Some teams find that this is not very useful for the developer or customizer to indicate what this relationship represents.

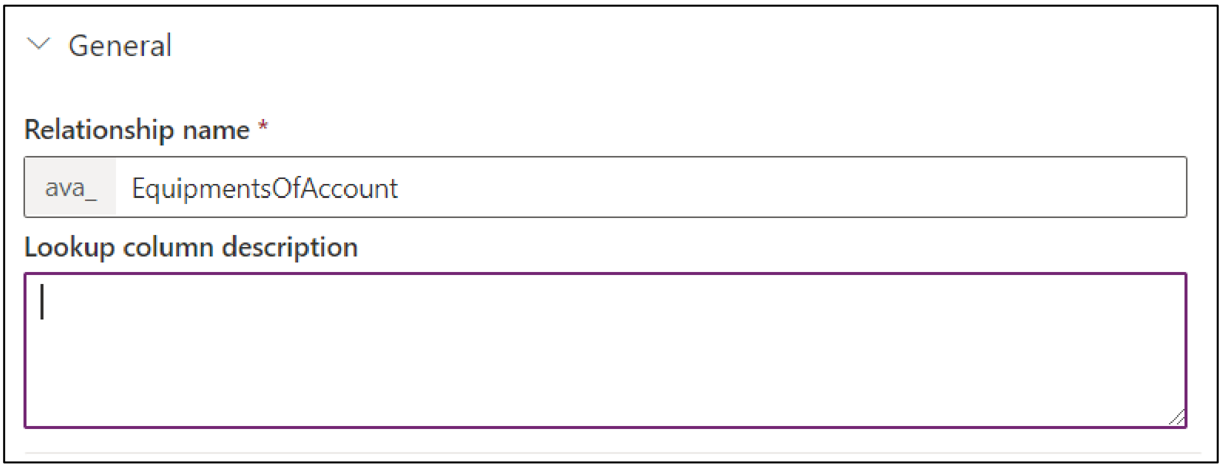


Figure 13 Relationship name

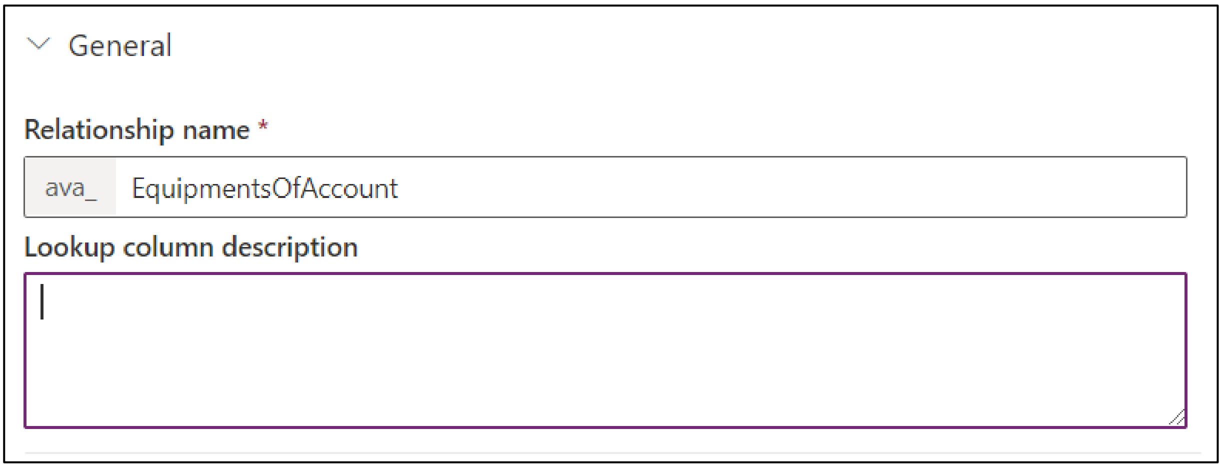
9.2.1 Option One - Rule 1 – Regular

1 – N Relationship:

Rules:

- **prefix_“PrimaryTable“+"Middleword"+“Plural RelatedEntity“**
- As Middleword it is useful when you use "Of" as a connector of the tables.
- The Table which is "N" should also contain the plural name of it. Add "s" at the end.
- Use **Pascal Casing** to make it clear in the schema name what is mentioned with this relation.

Example:
1 Area has N LBUs. One LBU has exact one Area in this case we will name the relationship „ava_AreaOfLbus"



9.2.2 Option Two - Rule 2 – More than One Relationship Between the Same Tables

Between the same relationships exists more than one relationship and each relationship describes a specific responsibility or relation. The better the relations are described the better the Project Team can navigate through the CRM and Code structure.

For example, when a country has one Service Station Lookup which identifies the responsible Service Station.
The relationship name should be “**ava_ResponsibleServiceStationOfCountries**”.

And a second Service Station Lookup which identifies the responsible Main Office.
The relationship should be "**ava_MainOfficeServiceStationOfCountries**".

In this case also change the Custom Label in the Display Options otherwise it will be shown with the default plural name in the navigation.

Rules:

- **prefix_“PrimaryTable“_+"Middleword"_+“Plural RelatedEntitiy“_ + “Foreign Key Name”**

9.2.3 Option Three

Communicate to the team the reasoning behind the naming coming from Microsoft. Explain that every field is tagged with the publisher prefix such as ava and that relationships are a combined naming of two schema names.

For example: ava_ava_CustomField1_ava_CustomField2
prefix_prefix_FieldName_prefix_FieldName

9.3 Display Options

Prior to configuring the relationship, the customizer should understand how the business would like this new field displayed. Should the field show in the navigation pane or only on the primary table form.

It might be that the field should only be used as part of a sub-grid. It is important to get these details aligned prior to configuration.

If it is not specified, then the display option default can be set to “Not Visible”.

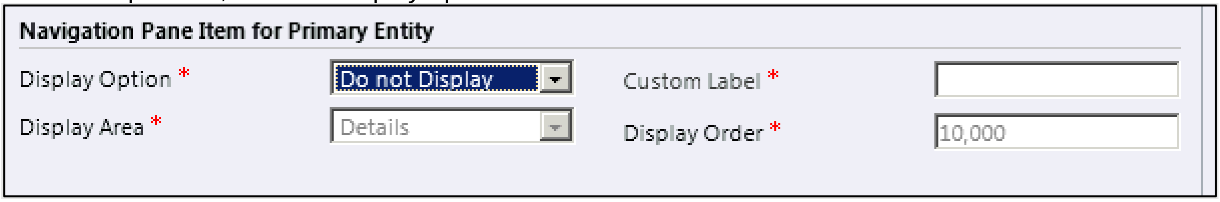


Figure 14 Display options of relations

9.4 Cascading

The architecture of the relationships is something that needs to be carefully considered, because these choices can impact what data updates child rows and what data gets deleted if a parental relationship exists.

If possible, the cascading setting must be validated and set to a specific requirement.

In reviewing established projects, the cascading for assign and deletion are extra critical and can be reviewed using the below sections.

> ***Note***
>
> This might not be possible during the initial customization so you will want to make sure to have a user story to review in a later sprint. Refactoring might be required.

We suggest validating the cascading setting for all relevant tables (also standard tables) before the entry of any data and before any data migration.

We also strongly advise that every new project check and double check the architecture of the parent relationships. Parent relationships should be challenged and understood.

> ***Note***
> The reason for the challenge is that the parent setting leads to the creation of implicit shares and this in turn leads to a growing Principal Object Access (POA) table which can be a performance killer.

9.4.1 Use Cascading Template: Parental

Use this, if the table is a clear sub table or child table of a master table.
For Example: quote header and quote line item.

1. It does not make sense to have a quote line item without a header. If the header will be deleted, then all line items that are part of that quote should be deleted.
2. The owner of the header will always be the owner of the line items
3. **Each table would only have ONE parent!**
4. **No change on behavior can be customized!**

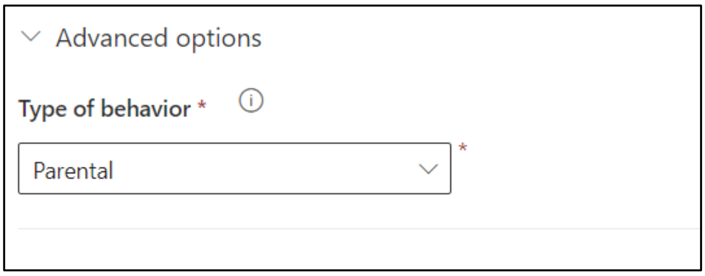


Figure 15 Parental behavior

> ***Important Note***
> Security privileges are not prohibiting Parental cascading of deletion! E.g. An Account A exists with multiple associated cases. Additionally, case entity is parental to accounts. User B does not have privilege to delete cases but accounts. He deletes account A then no error will come up. Cascading is done with elevated privileges and the privileges of current users are not checked!

9.4.2 Use Cascading Template: Referential

You can choose between delete is only possible if no links exist or the link will be removed but the row will not be deleted.

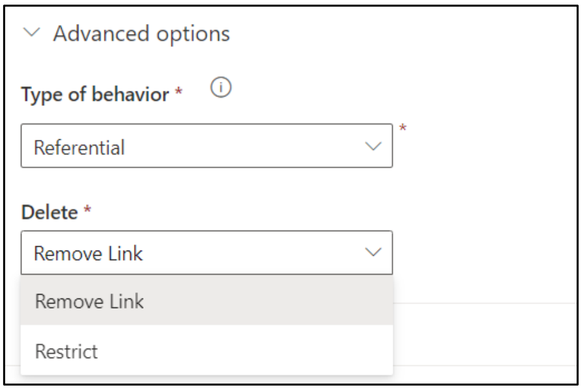


Figure 16 Referential Cascading

9.4.2.1 Delete Restrict

Use this behavior when the deletion should be restricted if associated rows do exist.

Example:
You should not be able to delete a payment term if used in any related quote.

9.4.2.2 Remove Link (standard if no decision)

If the restrict delete and an auto delete is not useful, and the tables are loosely coupled, use “Referential Remove Link”.

Example:
A Project could have many visit reports, the visit report also a link to the customer.
If the project will be deleted, the visit report should still exist on the customer history.

9.4.3 Use Cascading Template: Configurable Cascading

Cascade None, but for Delete Cascade All.

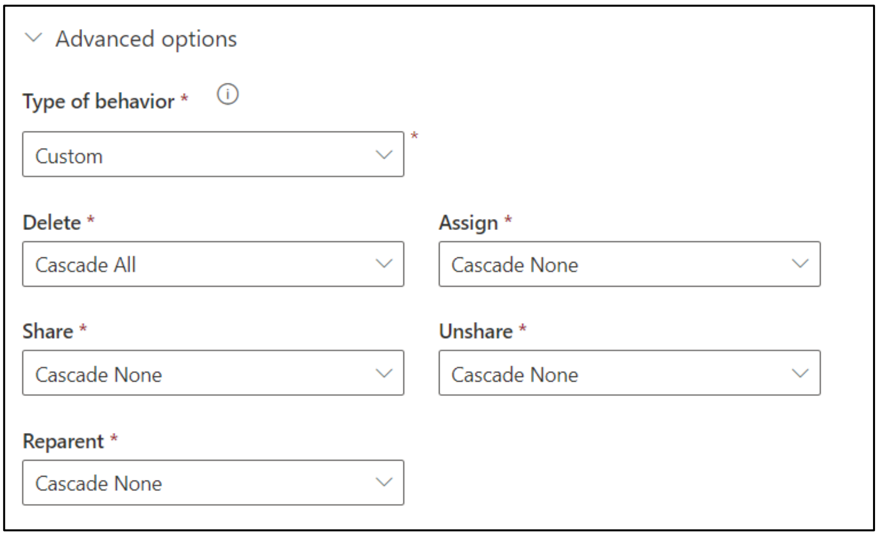


Figure 17 Cascade Custom

Choose if the row should be auto deleted but the parental setting is not useful.

9.5 Mapping

The default mapping generates only a mapping for the relationship lookup field. If the tables contain additional fields with the same content, the mapping should be extended. The mapping only works on the creation of a child row.

Note

The mapping can currently only be changed if you switch to the classic editor.

10 Creating New Columns

10.1 Naming

- **Use Pascal Casing for schema names!**
  e.g. *prefix_SerialNumber* and not *prefix_serialnumber*.
- Don’t create a column with the same schema name as an existing out of the box field.
- Don’t create a column with the same label name or change a label name of a column to an already existing one in the same table. The excel export/import will fail.
- Schema names must start and end with a character, it is forbidden to start or end with a number.

Design Decision
The reason is that for the column “ava_50el”, the code generation will generate 50el as this variable name is forbidden according to C# conventions.

A column with a number at the end is also bad, because if the field is more than once on the form, the system concatenates against the number as a suffix

Keep in mind that if you create new fields the **“Name” column** will always be shown in **lower case** even if you correctly used the Pascal Casing for the schema name.

Field Type	Schema Name format to apply	Name template	Example
Text / Decimal / Date / Currency	Capital letter for the first letter of each word	ava_XxxxxYyyy	ava_ClientName
Choice	Capital letter for the first letter of each word + suffix: Code	ava_XxxxxYyyyCode	ava_ClientCategoryCode
Choices (MultiSelect Choice)	Capital letter for the first letter of each word + suffix: Codes	ava_XxxxxYyyyCodes	ava_ClientCategoryCodes
Boolean (Two Options) *	Capital letter for the first letter of each word + prefix: is	ava_IsXxxxxYyy	ava_IsRegularClient
Lookup	Capital letter for the first letter of each word + suffix: Id	ava_XxxxxYyyId	ava_ClientId

Important Note
Do not use the same label name of a column more than once in the same table. CRM will work fine, but the excel export/import will fail, if this duplicate column will be included.

10.2 Description

- A precise description for a Column must be added when it is surfaced to an end-user and when the description (which rendered as a tooltip on the form UI) adds valuable guidance for the user.
- Descriptions must not start with the field name otherwise the tool tip will have duplicated text.
- Descriptions should be ended with a full stop “.”.
- *Examples:*

- If the column value can be directly entered by a user, follow the Microsoft wording, for example "Type the company or business name.".
- If the column value is filled by a backend system, the wording can be for example "Stores the SAP number.".

10.3 Changing the Length on Text Fields (nvarchar)

The length can be changed after the field is created, but be aware of the following behavior:

If the field length is changed to a higher number and later this number is decreased the decrease is not saved in the database.

Therefore, all the old data will still be saved in the field and not cut! However, the next time a user opens the record and saves it, the field will be truncated at that time.

For examples:

Field Type	Length	Field Content
Text	30	1234567890
Text	5	1234567890
Text	6	1234567890

10.4 Creating Choices

- **Local vs. Global choices:** If there is any chance that your choice field is being reused in multiple tables with the same choice values now or in the future (for example generic fields like a "Type"), it is an indication that you should use a global choice set. Otherwise, use a local choice set.

	Local choice set	Global choice set
Usage	If values are not shared by more than one table type or the values are not the same for each table type.	If the values should be the same, shared across multiple table types or multiple columns. (This helps especially with mappings in backend code or integrations.)
Behavior	Each table has its own set of values. No "sharing" of the values takes place.	Each value is available on all tables. If you change or add a value you change it for every table.
Cons	No sharing of values each value must be entered manually.	No good possibilities to hide specific values on specific tables or columns.

- Leave the enumeration as it is. You don't need to enumerate them with value 0, 1, 2, 3... This is just nice looking, and it will save you efforts later.
  Exception: If the value is used in an interface, then change it to the correct number.

> ***Note***
> *You don't need to add "Code" at the end of the technical name (as it is not a field).*

10.4.1 Choice vs. Lookup vs. Choices

To realize a selection list, there are three different alternatives:

- Choice
- Choices
- Lookup

Choices have nearly the same behavior as the Choice Set besides you can select multiple values. Differences are:

- No default value can be set
- Processes/Workflows: Choices sets can be used as trigger events for Processes/workflows, but they cannot be updated within the process/workflow logic, nor can they be used in a conditional statement to perform some other branching logic condition.
- Business Rules: Choices cannot be updated within the eue logic, nor can they be used in a conditional statement to perform some other branching logic condition.
- Larger lists are possible and supported via search capability but still not recommended for high volume lists.

To choose, use the following considerations:

Consideration	Choice / Choices	Lookup
Additional Attributes needed	Not possible. You have Value, Label and Description	If the list should contain further information (not only a name) you should always use a lookup. E.g. the country list, should also display an ISO code, and a risk classification of the country.
Control program logic	If the list must control a program logic, and Choice is the easiest choice. E.g. if value “a” is selected tab “b” should be hidden Because the value of the Choice is on all systems the same, and it is not possible to change it without customizing changes.	If not handled specifically, each lookup list entry will get a unique GUID which is different in each environment. While this could speak against the use of lookups, there is a relatively simple way to overcome this, namely by defining the same GUID for each row during import (for example by using the CRM EMC tool and having a central Excel defining this set of rows including GUIDs).
Data volume (usability)	Good for small lists (2-15) rows, search is available for larger choices list but not recommended.	Good for huge lists > 15 rows Search dialog exists User can sort and filter the list
Change Management	The list is stored in the customization. Adding / Removing entries must be done in customizations If the customer business departments should maintain the content of the list and the content of the list is changing often the lookup is a better alternative. If the customer wants to have more control over the values option sets is the way to go forward.	The list content is just data. User with security privileges could change it, which is a huge benefit, especially in larger systems where you want to enable key users to manage data themselves without relying on developers or the 3rd level support.
Ability to filter rows	Filtering choices list based on other form content is only possibly by JScript development.	Lookups could be filtered by customization.

Consideration	Choice / Choices	Lookup
Security	Every user sees any list entry. No security settings are possible	It could be configured for user to see only special list content by security e.g. on opportunity I could only select the customers of my business unit
Offline Scenarios	Per default offline	Must be set to offline available and the rows must be synchronized
Multi language	Has multi language support by default, but the translation could only be done via Metadata (customizing) no local administration.	No default multi language support via standard customization. If multi language is needed, this must be implemented, by „Flexi“ or „Multi Language“ concept
Deployment	The content of the choice list will be deployed always by customizing.	The content of the lookup table must be deployed manually by csv ex- and import to all systems
Multi Selection	Is possible via Choices selection lists	Lookups cannot be used for multi selection, but with sub grids (n-m, n-1 relation) you could build this. Another version to build multi pick list is to customize checkboxes for every list row
Rapid, Frequent changes	Hide of rows can only be done via implementation (you must hide the Option Set values via JScript).	Hide rows can be done by filtering and deactivation of the row.

10.5 Excel import as the source of customization

The standard excel import can create additional columns.

Important Note

Be aware depending on the import you can add columns from an import file as new columns in a table. This behavior is not desired.

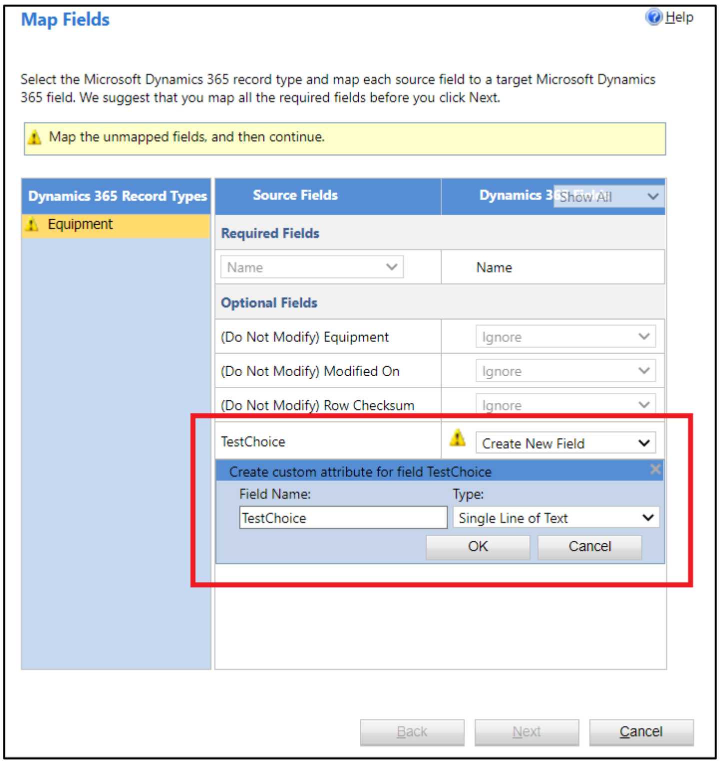


Figure 18 Customization via excel import

11 Creating Forms

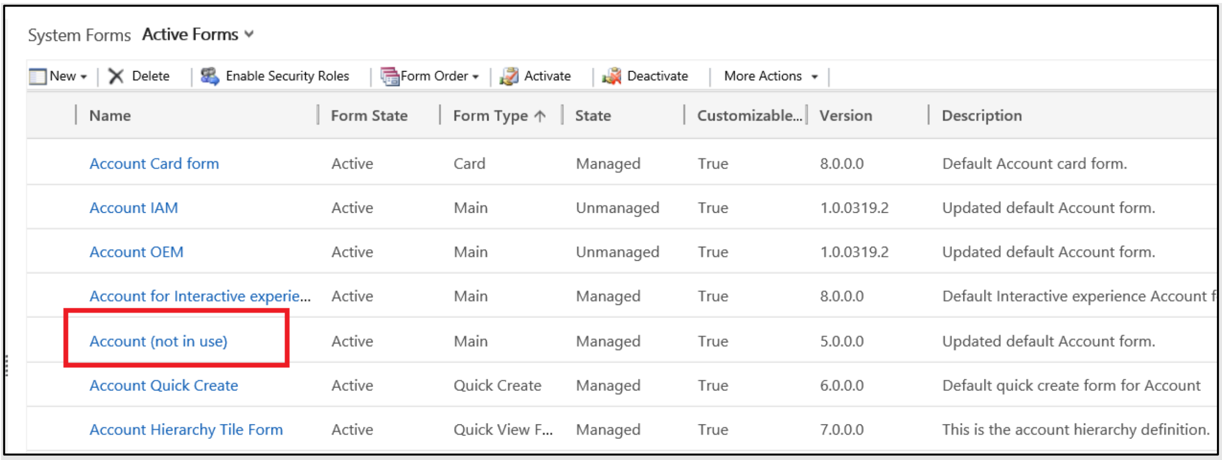
11.1 Considerations

- Create only the forms necessary.
- Forms can be app and/or security role specific so you can create many forms, but each will need to be maintained.
- Change the schema names of tabs, sections, and sub-grids. Use pascal casing for schema names, for examples:
    - tabSerialNumber
    - secSerialNumber/sectionSerialNumber
    - subgActivities/subgridActivities
  - This makes it easier for developers to find and work with the sections and tabs.
- If JavaScript form dependent code is generated, deletion or movement of sections and fields should induct re-generation of the JavaScript code.
- Do not forget to add the newly created forms to the corresponding Apps.

> ***Recommendation***
> On Quote, Opportunity, and Marketing List, restrain from changing schema names of existing components on the form. Microsoft standard logic uses these fields in their implementation. Recreating them won't work on most occasions.

Copy forms of standard tables if you are customizing them.

You can label the OOB forms not used, but better to leave them as they are so that any weekly update from Microsoft does not find something unexpected.


System Forms Active Forms

New | Delete | Enable Security Roles | Form Order | Activate | Deactivate | More Actions

Name	Form State	Form Type ↑	State	Customizable...	Version	Description
Account Card form	Active	Card	Managed	True	8.0.0.0	Default Account card form.
Account IAM	Active	Main	Unmanaged	True	1.0.0319.2	Updated default Account form.
Account OEM	Active	Main	Unmanaged	True	1.0.0319.2	Updated default Account form.
Account for Interactive experie...	Active	Main	Managed	True	8.0.0.0	Default Interactive experience Account f
Account (not in use)	Active	Main	Managed	True	5.0.0.0	Updated default Account form.
Account Quick Create	Active	Quick Create	Managed	True	6.0.0.0	Default quick create form for Account
Account Hierarchy Tile Form	Active	Quick View F...	Managed	True	7.0.0.0	This is the account hierarchy definition.

Figure 19 Forms list

11.2 Form Display

Forms are displayed according to the visibility of the form and the form order. If no form is available for a specific security role, the Fallback Form is displayed. If possible, only 1 fallback form should be available for each table.

To set the fallback form Click on "Enable Security Roles".

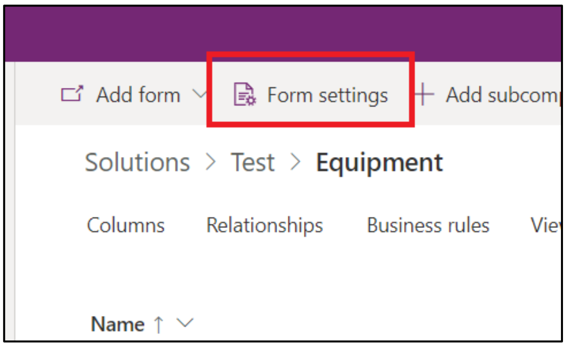


Figure 20 Navigate to Fallback Form Step 1

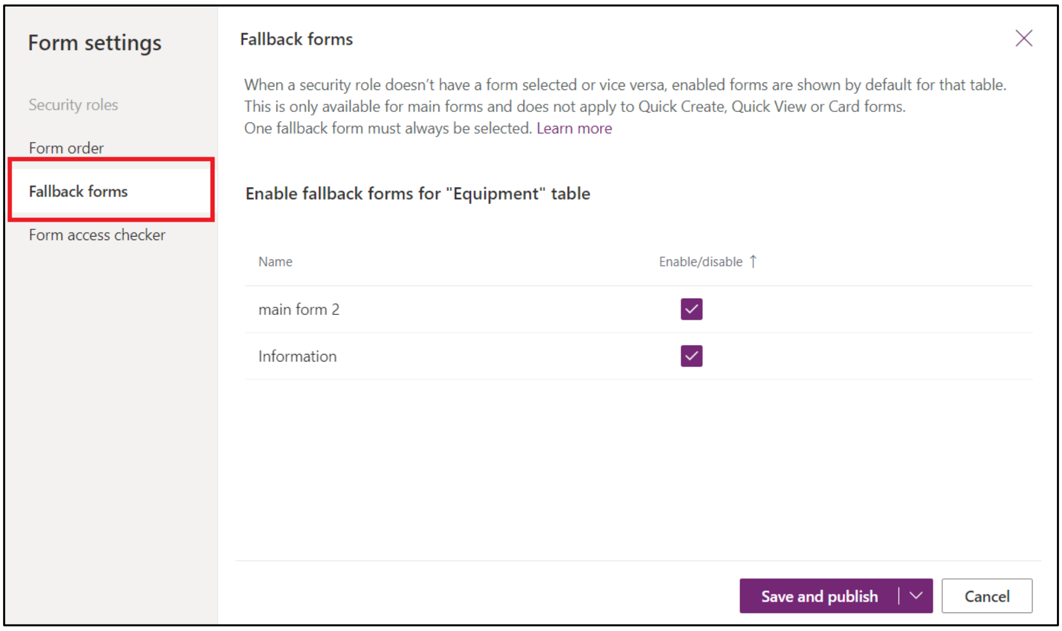


Figure 21 Navigate to Fallback forms Step 2

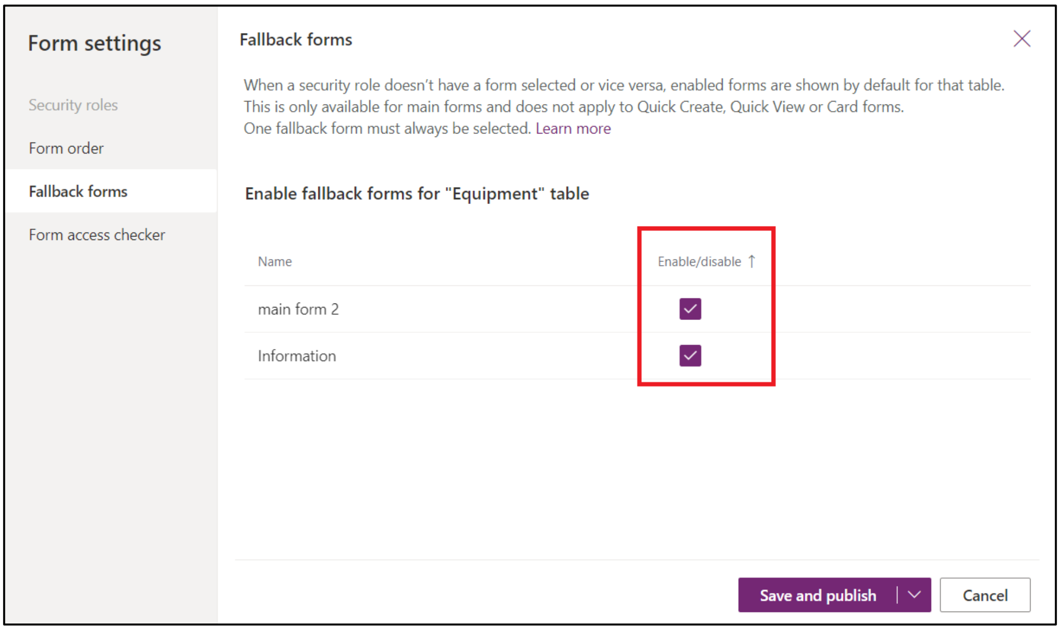


Figure 22 Set Fallback Form

11.3 Multiple Forms

You can easily use multiple forms with different layouts and fields present on each of the forms. Keep in mind that for every form, there could be a different ribbon and or JavaScript.

If your JavaScript implementation is based on columns which are not available on every form this might cause an issue.

12 Views

12.1 Main Considerations

- Always leave the primary column of the table in the view.
- Customize as few views as possible.
-

Design Decision
Change the layout of all views to the appropriate one. If not discussed with the customer use the View Layout Replicator plugin of the XrmToolbox to copy the view layout to all the views on a table.

- Once the views are copied, change the associated views and lookup views as needed.
- Always configure the inactive and search views of a table, especially if the table was newly created!
- Views consists of columns, but they also consist of corresponding filters and sorting!
- No special naming conventions for view names currently exist, but the names need to be clean, and consistent across tables and they sort alphanumerically in the view list.
- Deactivate views that are not in use and remove them from your app(s), but keep in mind deactivation is not transferred via solution import/export so you might need to do this manually.

12.2 Follow Views

- Follow views cannot be exported via a solution, therefore it is no use to document potential changes to the Follow views in the Table Views workbook, because even though they could be done in a development environment, they could not be moved to another environment or stored in Azure DevOps
- Do not include follow views in any design work.

13 Dashboards

There are different types of dashboards. Use the type of dashboard to meet the requirement.

1. Classic Dashboards
2. Interactive Dashboards – Single Stream
3. Interactive Dashboards – Multi-Stream
4. Interactive, Table specific dashboards

Recommendation

Keep in mind that the default layouts can be changed! You can easily add more than 6 sections to one dashboard. But it’s not recommended because of loading times.

14 Apps

Apps are used to setup a specific range of tables and views and dashboards for a specific user group. This allows you to tailor your solution to specific user requirements.

Dynamics 365 are a set of Model-driven apps.
To create one go to the Maker Portal, Select, or create a solution and create your app.
Select “+New” >> “App” >> “Model-driven app”.

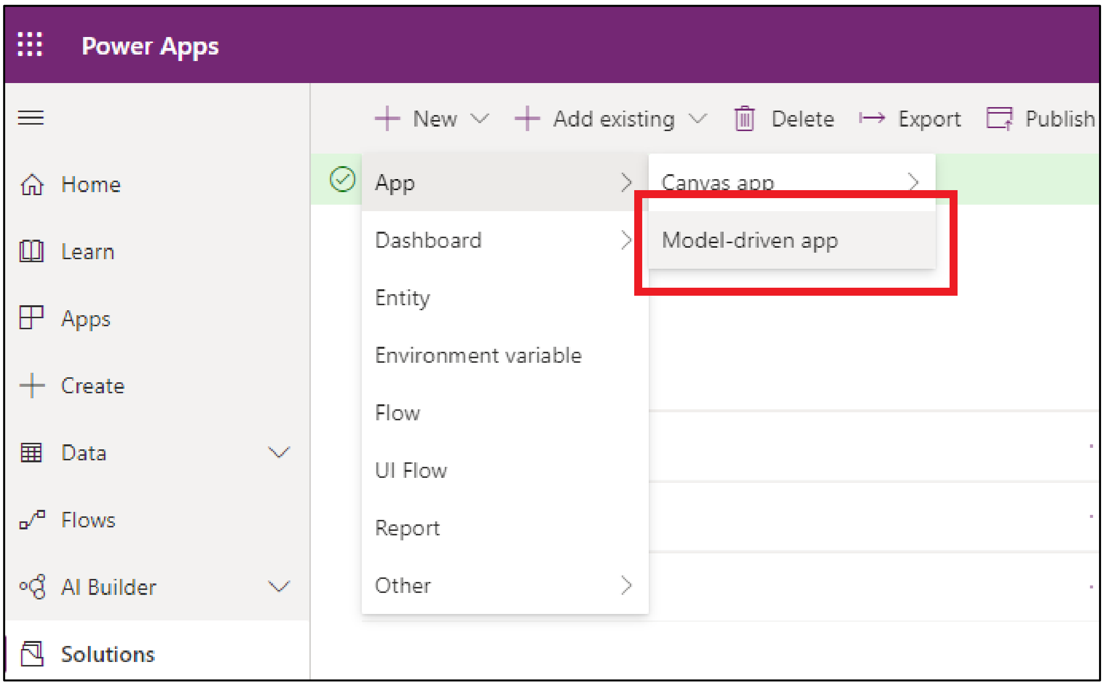


Figure 23 Create a Model-driven app

Apps are the only way of using the new UI “unified interface” (introduced in CRM 9.0).

- You can create Apps and these apps can be security role specific.
- An App consists of a specific navigation
- You can select (hide) specific views, dashboards, and forms in the app.
- An App can be transferred via solutions including the security role mapping.
- The App designer allows you to customize navigation, views etc.
- Give the app a distinguishable name and description. You can change the logo and use a welcome page for displaying app specific data.
- Be aware that only Unified Interface apps will be available for the mobile client!

Note
Do not customize the standard Field service or Sales App! Create a new one and tailor this App to the requirements of the customer

Important Note
There are specific apps which cannot be copied. Additionally, Microsoft is insisting on using the standard app for Field Service. Clarify this in your project based on the current behavior of the Power Platform!

You don't need to start from scratch you can use an existing Sitemap from an existing App as the start. Also do not use the Default app! This app is still of type Web and not Unified Interface.

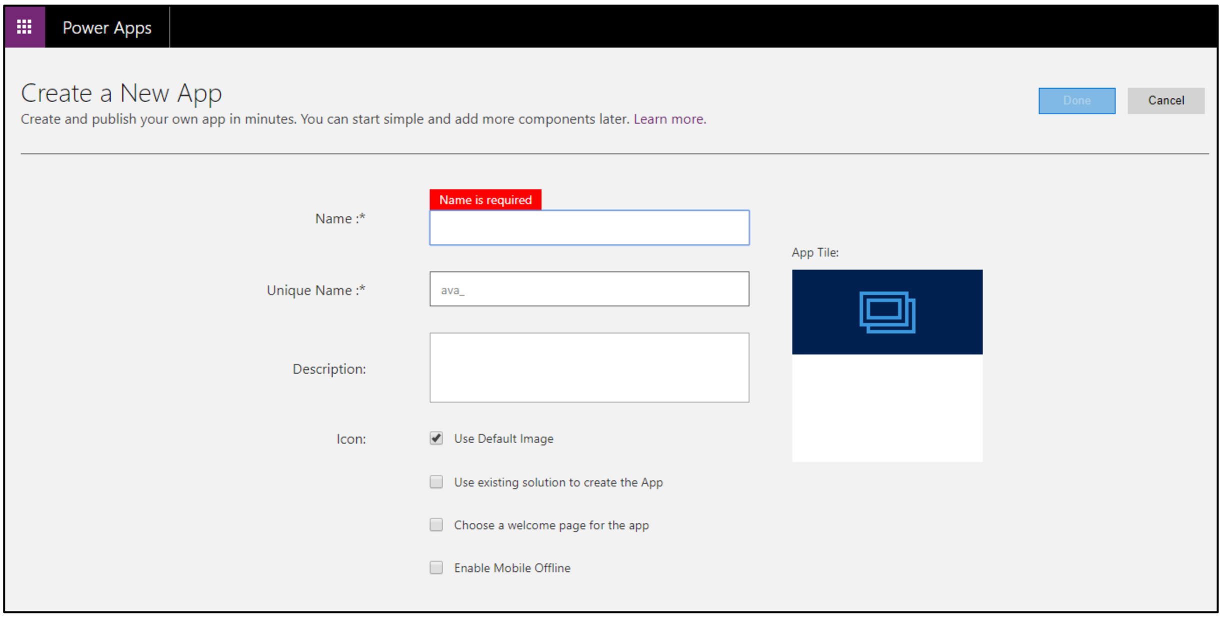


Figure 24 App Designer

If you choose "Use an existing solution to create the App" you must select a solution and then afterwards the sitemap which is in this solution.

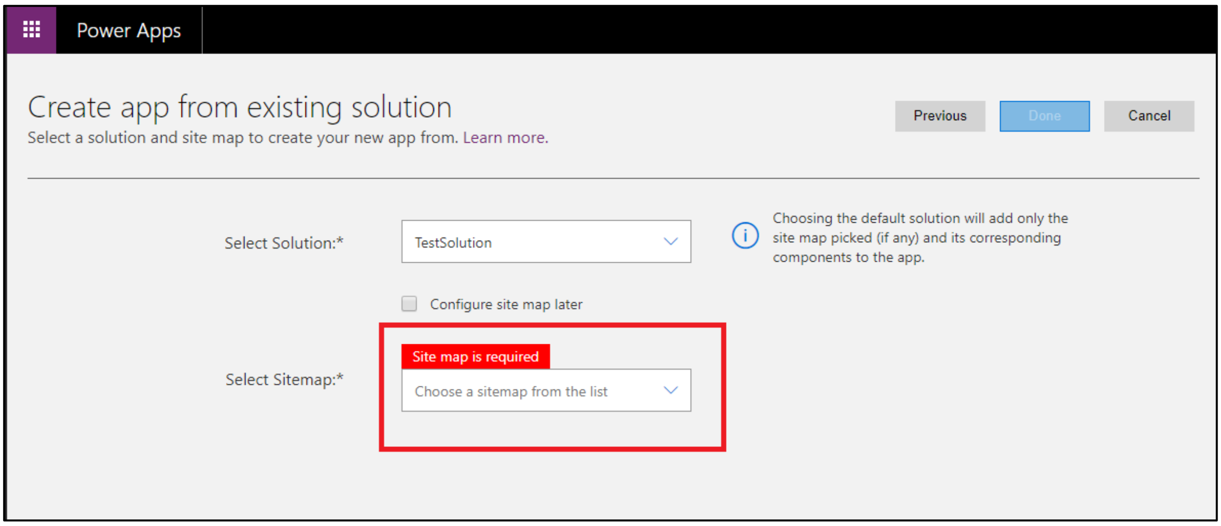


Figure 25 Choose existing Sitemap

You can create an App without using the sitemap then only the tables which are in the solution will be added to the App, but the sitemap will be empty.

Recommendation
Use existing solution to create the App" means that you can choose a solution and based on the sitemaps in this solution the app can be created for you.

14.1 Sitemap Customizing

For each app a specific sitemap is created. You can customize the sitemap via the Sitemap Designer.

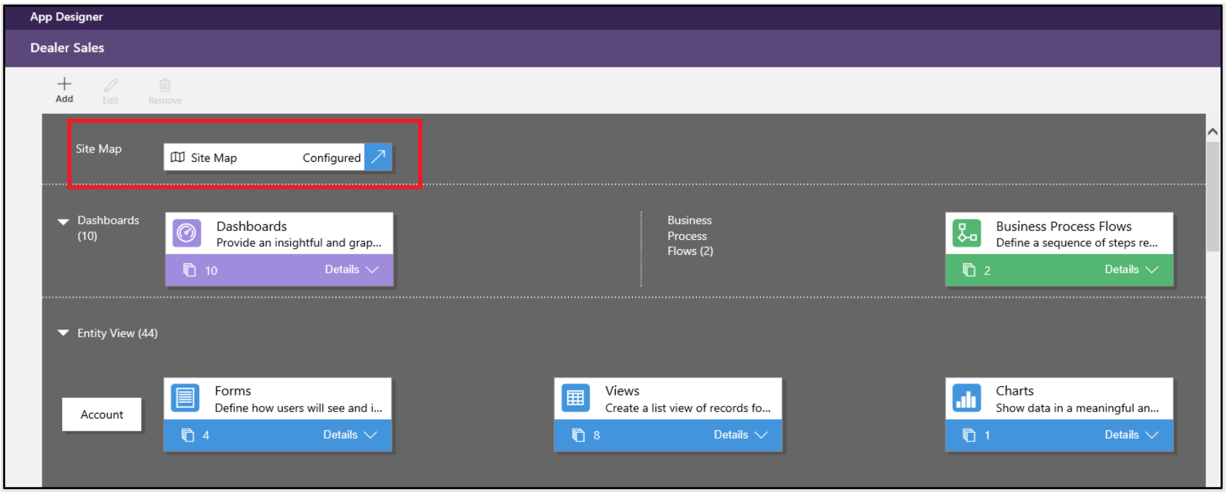


Figure 26 App Designer Sitemap Customization

Open the Sitemap Designer there you change and add Areas, Groups and Subareas.

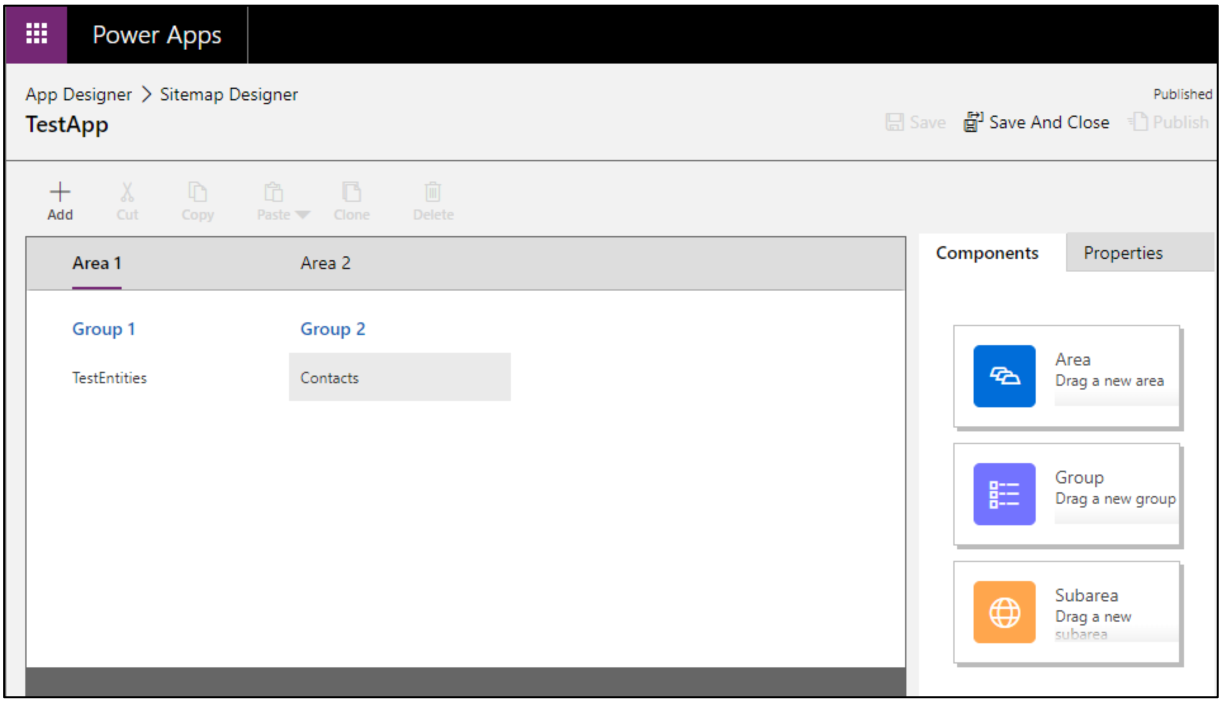


Figure 27 Sitemap Designer

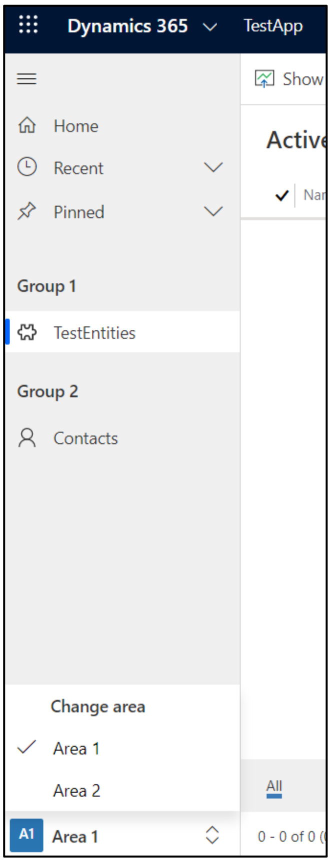
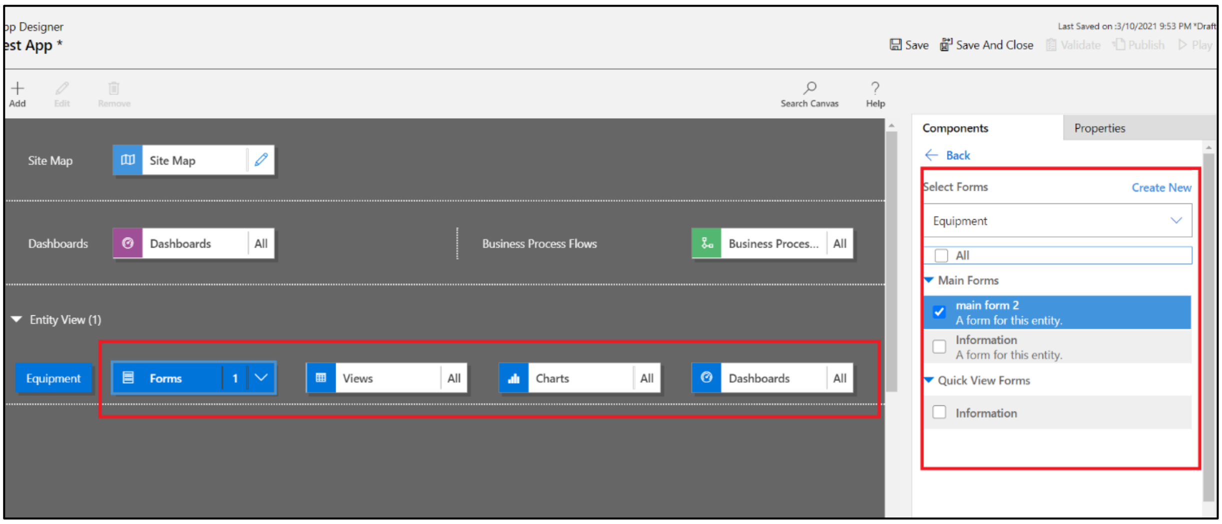


Figure 28 Sitemap as the user will see it

Areas are selected via the button and change the complete navigation.
Groups are for grouping specific navigation items. They will not be displayed if no Subarea is created in the app.

14.2 Forms, Views, Charts, Dashboard selection

For each Table you can select which Forms, Views, Charts and Dashboards should be available for the users if they choose this app.



In this way you can restrict the access to these types regardless of the security roles the user has.

Important Note

If you restrict the access any of these types for a table and you later add another form, dashboard, chart or view you must manually make it available in your app! It will not be added automatically to your app!

If you do not restrict access and leave “All” selected each of your customized form, view, chart, or dashboard will be made available in the app automatically.

14.3 Canvas Apps

There is a wealth of guidance available for the configuration of Canvas Apps, please refer to the Power Platform Community for the respective guidelines for naming conventions etc.

With regards to embedding of Canvas App into model-driven, consider the following topics among others:

- The look and feel should be similar, therefore a default theme should be chosen, for example “Office Blue” can work ok.
- Think about scaling factors and font sizes from the beginning to avoid spending a lot of efforts at the end.
- Create an AD group to share the app (ideally the same group can be used as for the organization access but as of 12/2021 this does not work yet).

14.4 Outlook App

Guidelines have to be added.

15 Ribbon Button Customizations

15.1 Main Considerations

Use the Ribbon Workbench Plugin for ribbon customizations. The tool is part of the XrmToolbox see 29.1 Tools: XrmToolbox.

- If you must hide buttons based on security privileges, then create a Table and hide the buttons based on security rights on this table and name this table “secTablename” or similar. The prefix shows this is a security table.
- Be aware the ribbon is part of the table customizations. That means if you customize a ribbon you basically “download” this table in the corresponding tool.
- You should not customize the table and the ribbon in parallel because the ribbon customizations will overwrite your “other” customizations!
- IDs must be unique and should be self-explanatory.

Important Note

The Command Bar Editor has been introduced by Microsoft (preview as of 12/2021) and could at some point replace the Ribbon Workbench.

15.2 Buttons

- The button id should be self-explanatory therefore remove all not needed prefixes.
  For example: “ava.productlineitem.AddIncentives.Button”
- You must add a command to a button otherwise the button will not have any function.

15.3 Commands

- A Command defines the actual “action” of the button and contains Display and Enable rules.
- Command Ids should be self-explanatory and should contain the name of the button for which the command was created.
  For example: ava.productlineitem.AddIncentives.Command

15.4 Display & Enable Rules

- The Id should contain the reason and if applicable describe the action itself. For example: IfAccountHasPositiveRevenue, IfUserIsSetAsXXX)

16 Icons

Icons can be used to brand a delivery and to better communicate and distinguish tables and Apps. SVG icons should be used for all Unified Interface systems, they resize without losing their display quality.

The icon file size of SVG must be less than 10 KB.

16.1 Naming

Schema
ava_/images/<ribbon|global|sitemap>/<table logical name (optional)>_purpose_.svg

Example
ava_/images/ribbon/upload_.svg

17 D365 Security

All Dynamics 365 security requirements must be part of the technical and functional architectural design at the start of the project and reviewed throughout any engagement. It also needs to be part of the performance architecture of the system.
This document will not cover the decision process and questions but will give guidance how to customize specific parts of it.

There are many starting points for consideration.

	Important Note The 360° View of data is one of the key concepts that clients request and need, security empowers who can see what. It's not meant to hide data!

- Security Automation

17.1 Security of Environments with and without Dataverse

An environment without a Dataverse does not provide the capabilities which are discussed here.

Environments with Dataverse add support for more advanced security models that are specific to controlling access to data and services in the Dataverse environment.

	Note The default environment is an environment without a Dataverse. A Dataverse is a requirement of any D365 App therefore the advanced security models do apply here.

17.2 Dynamics 365 Security Architecture

Parts of the D365 security architecture are:

- Security Roles
- Access Teams
- Hierarchical Security
- Business Unit Configuration to support security
- Column Level Security
- Encryption

17.3 Security Roles

Security roles are a specific group of security privileges and define how different users access different types of rows.
Each user can have multiple security roles.

Security role privileges are cumulative: having more than one security role gives a user every privilege available in every role.

Each security role consists of row-level privileges and task-based privileges.

Row-level privileges define which tasks a user with access to the row can do, such as Read, Create, Delete, Write, Assign, Share, Append, and Append To. Append means to attach another row, such as an activity or note, to a row. Append to means to be attached to a row.

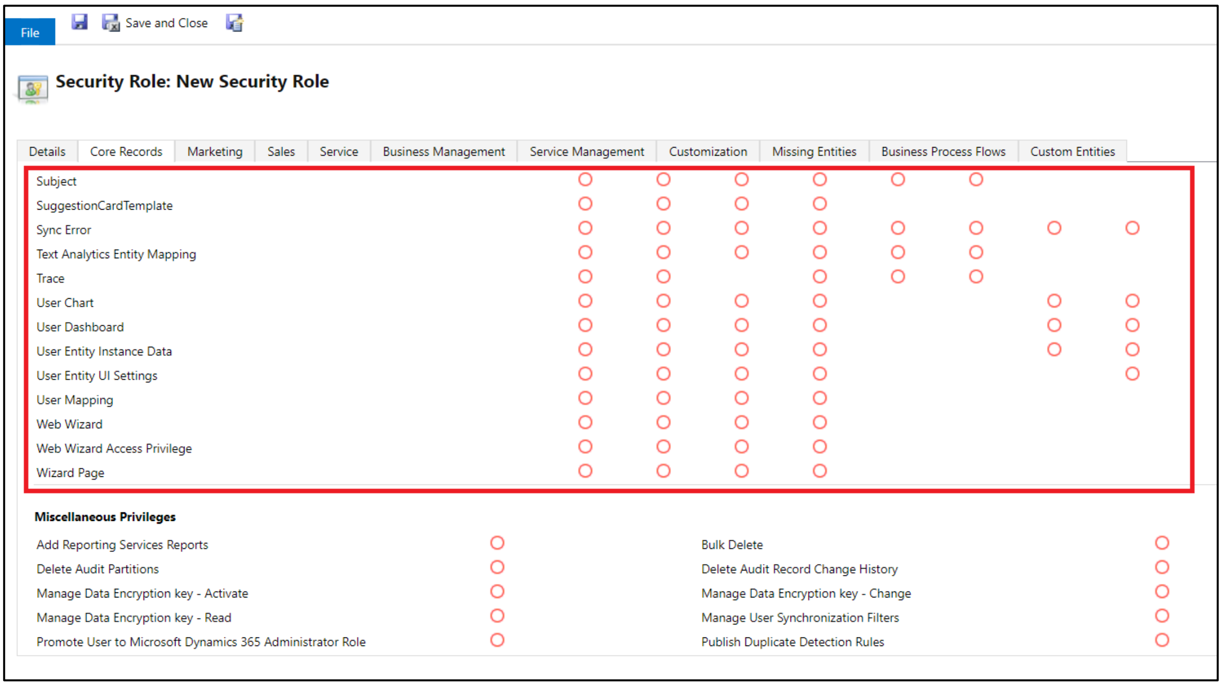


Figure 29 Row based privileges

Task-based privileges, at the bottom of the form, give a user privileges to perform specific tasks, such as export to excel.

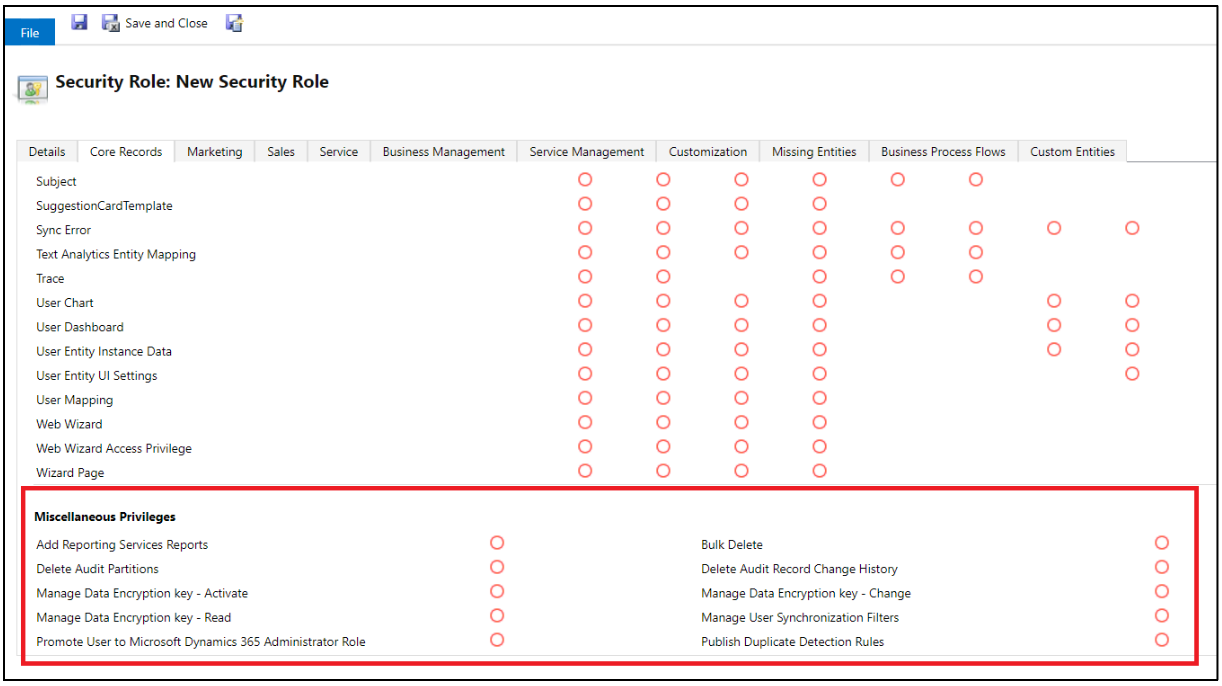


Figure 30 Task based privileges

Each security role that is delivered with any of the out of the box app, might have security privileges configured that you potentially can’t see.

	Important Note To setup project specific security roles, clone an existing role and configure.

- Prefix them with 00, 10, etc. or use a customer prefix so they appear first.
- **Never delete any security role, leave the default ones untouched.**

	Important Note Security roles can be applied directly to a user or indirectly via team membership. Each user should have at least one directly assigned security role! If no direct security role is assigned to a user, you will have issues. E.g. excel import will not work and other.

Icon	Description
	Organization (Global) - This access level gives a user access to all rows in the organization, regardless of the business unit hierarchical level that the environment or the user belongs to. Users who have Global access automatically have Deep, Local, and Basic access, also.
	Parent: Child Business Units (Deep) - This access level gives a user access to rows in the user's business unit and all business units subordinate to the user's business unit. Users who have Deep access automatically have Local and Basic access, also.
	Business Unit (Local) - This access level gives a user access to rows in the user's business unit. Users who have Local access automatically have Basic access, also.
	User (Basic) - This access level gives a user access to rows that the user owns, objects that are shared with the user, and objects that are shared with a team that the user is a member of.
	None Selected - No access is allowed.

	Important Note You should restrict the amount of security roles a specific user has to max. 10 – 15. The reason here is performance.

17.3.1 Security Role member's privilege inheritance

Security Roles can have two possible privilege inheritance.

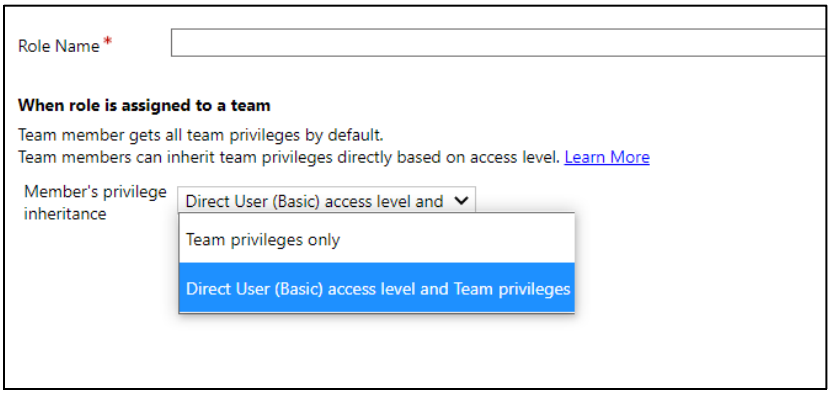


Figure 31 Security Role privilege inheritance

17.3.1.1 User privileges:

User is granted these privileges directly when a security role is assigned to the user. User can create and has access to rows created/owned by the user when Basic access level for Create and Read were given. This is the default setting for new security roles.

17.3.1.2 Team privileges:

User is granted these privileges as member of the team. For team members who do not have user privileges of their own, they can only create rows with the team as the owner and they have access to rows owned by the Team when Basic access level for Create and Read were given.

17.4 Column Security Profiles

- Column security profiles are a collection of these Field Permissions:
    - Read. Read-only access to the field's data.
    - Create. Users or teams in this profile can add data to this field when creating a row.
    - Update. Users or teams in this profile can update the field's data after it has been created.
- Give the Column Security Profiles a distinguishable name which describes for what this profile will be used.

Important Note
If you customize a Column to have column security be aware that from then one beside admins no user will even have read access. Therefore, create the basic needed field level security profiles right away.

Important Note
Column security profiles should be part of the customizing solution! The reason is that otherwise you will have dependency issues.

Important Note
Column security privileges apply to all rows of this table! You can not specify access to the column based on specific values of a row. E.g. give read access to an account number only if the account city begins with an “g”.

17.5 Hierarchical Security

Two types of hierarchical security are available:

17.5.1 Manager hierarchy

The hierarchy access only takes place if Hierarchy Security management is enabled in that organization and for that table, and if the user is a manager.

In this case, the user would have access to the row if both of the following are met:

The manager has a security role assigned directly or through a team that has the access level Business Unit or Parent: Child Business Units.

Plus, any one of the following:

- The row is owned by a direct report.
- A direct report is a member of the owner team.
- The row was shared to perform the required action with a direct report.
- The row was shared to perform the required action with a team a direct report belongs to.

17.5.2 Position hierarchy

Please refer to https://docs.microsoft.com/en-us/power-platform/admin/hierarchy-security#position-hierarchy, specific Avanade guidance will have to be added if required.

18 Business Process Flows

- Create Business Process Flows by starting in a solution.

> ***Important Note***
> Flows created outside of a solution and added to a solution are different than flows that are first created in a solution. Otherwise the entity which will be created will not have the correct prefix attached.

- Copying a Business Process Flow always creates it with a GUID
- Use <prefix>Bpf<BusinessProcessName >
- E.g. ava_BpfOpportunitySalesProcess

19 Cloud Flows

Cloud flows are an integral part of the Power Platform and give a wide range of capabilities. As of now there is no supported connector for Cloud Flows for On-Premise D365 environments available from Microsoft.

Do not use the legacy connector for “Selected Records” but a Canvas App instead.

Important Note
Creation of cloud flows should start in a solution otherwise the “current environment” triggers and some actions will not be available!

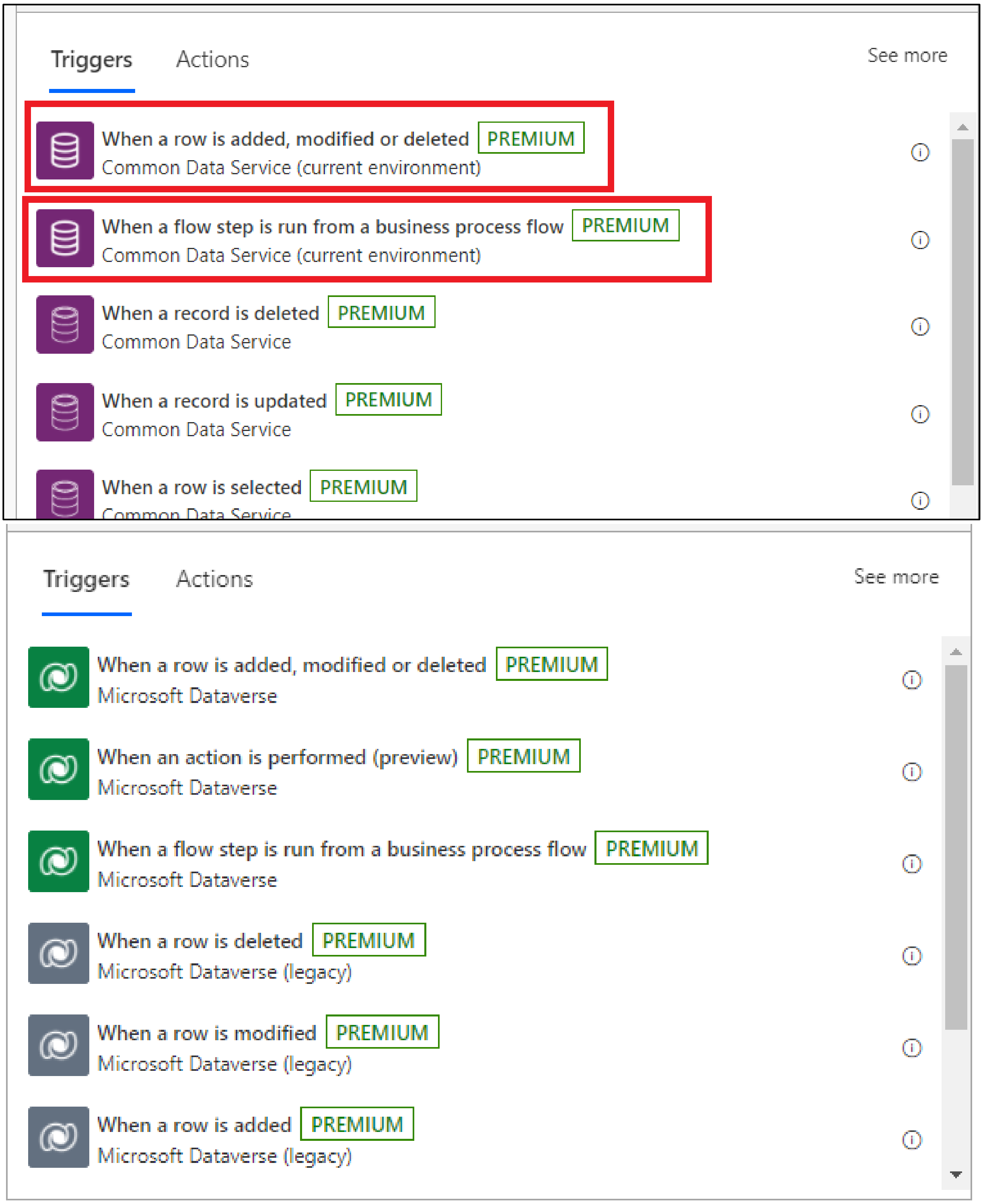


Figure 32 Flow current environment

You could still use not “current environment” flows but then you must specify the connection. The benefit of current environment triggers and actions are that flow connections are automatically created to the current environment where the flow is imported into.

19.1 Flow and Flow action labels

avanade

Design Decision
Rename the flow and each flow action so that the action has a distinguishable name! These names are used later in the editor in the flow.

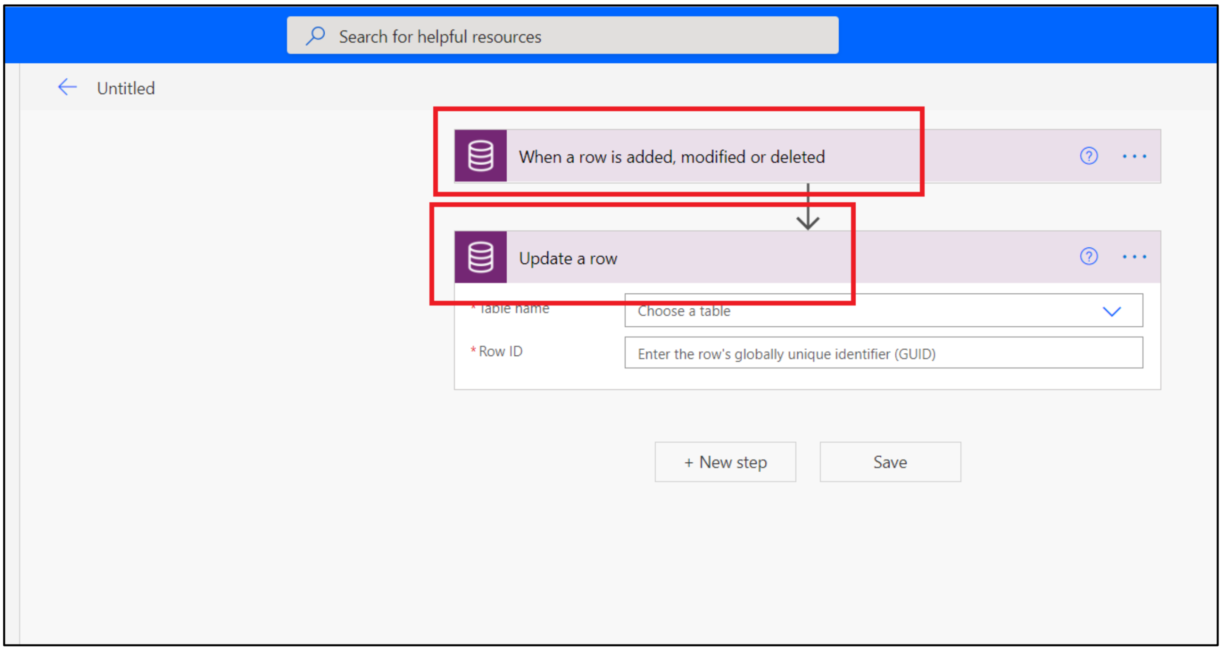


Figure 33 Rename flow actions

Example:

- Bad - Label: When a row is added, modified, or deleted
- Preferred - Label: On Create of Account when account name begins with G

- Bad - Label: Update a row
- Preferred - Label: Update Account set account status

19.2 Variable names

Use **pascal casing** for variable names (capital letter for the first letter of each word). Give the variable a good distinguishable label.

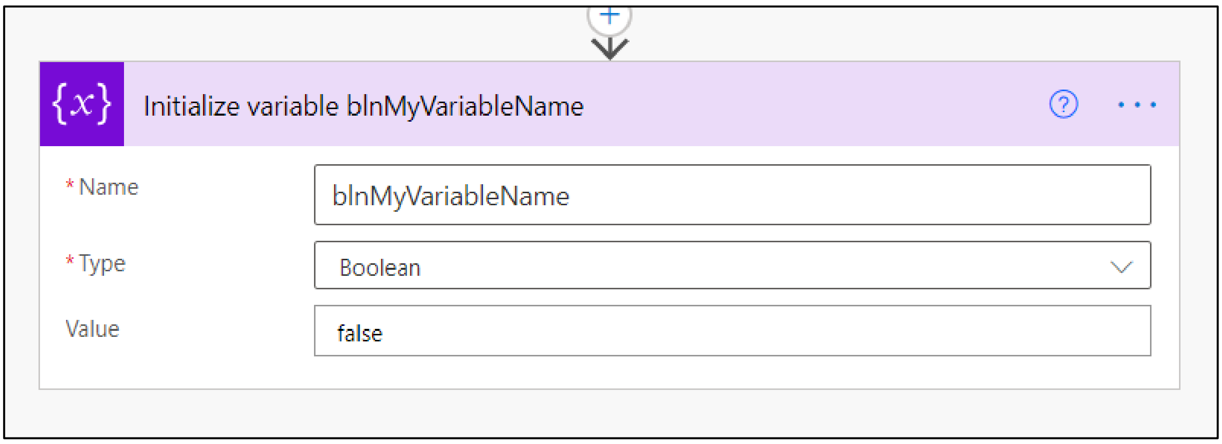


Figure 34 flow variable

Note
You will reference the variable via the variable name! But you should still rename the flow action of the variable initialization!

19.3 Visibility of flows

Cloud flows which are created inside a solution will only be accessible via the solution. You will not find them via the flow navigation.

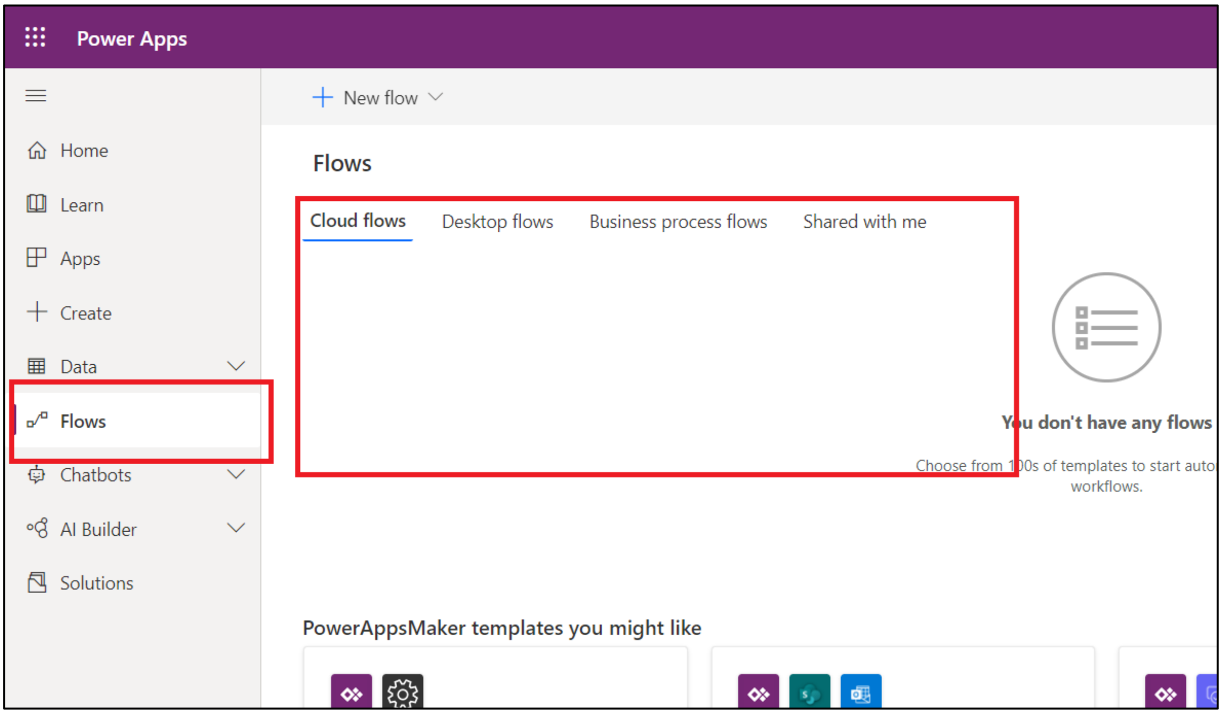


Figure 35 Cloud Flow visibility

For customization of the flows the same applies as for tables and columns.

20 Plugins

There are many considerations when building plugins within Dynamics 365. This section is just a short list of guidelines, but there are many more.

See the Development Guidelines in the BizApps Core Accelerator Wiki.

20.1 Server Messages

There are several server messages that developers can register plugins on that can cause performance issues if not done properly architected. These areas include

- Retrieve
- RetrieveMultiple
- RetrievePersonalWall
- RetrieveRecordWall
- RetrieveExchangeRate

The above will cause a plugin to be fired very frequently.

There are valid reasons for doing this though, but they are very niche and require the customer to understand and sign off on the performance penalties.

1. Custom security, far beyond what the product offers through security roles, field level security, row ownership, etc.
2. Localization support for individual rows, at runtime

20.2 API Limits

Microsoft introduced API limits into the product around 2018. This commonly presents an issue during a go-live event when the data migration scripts have been assumed to run under a single account, and the product starts sending back exceptions rather than importing rows.

The API limits are documented here and vary by product and license type:
https://docs.microsoft.com/en-ca/power-platform/admin/api-request-limits-allocations

There are two approaches for resolving this issue.

Option 1
If this is a go-live event and the product is hosted by Microsoft, open a Premier ticket or work with the customer's Fast Track engineer (if possible) to explain the situation. They have made exceptions in the past for customers so that the one-time migration can complete.

Option 2
The limits are based on the current user. If the migration is parallelized across multiple accounts and executed from different machines, this likely will not trigger the API limits.

21 Classic Workflows

The usage of Classic Workflows versus custom plug-in logic or outsourcing of any load via the Azure Service Bus or similar concepts is an important part of the architectural decisions. This chapter is therefore only relevant if the usage of Classic Workflows is applicable.

21.1 Scope

In general, the workflow scope shall be set to “Organization”, unless for some (rare) reason the workflow would be designed and configured to solely run in the User, Business Unit or Parent Child Business Unit scope.

21.2 Workflow Job Retention

The setting “Automatically delete completed workflow jobs (to save disk space)” must be checked, otherwise the corresponding DB Table gets too big and inserts take a long time so the overall Workflow performance will degrade.
If a customer wants to leave the workflows for a specific amount of time. Configure one or more Bulk Deletion Processes which are deleting these workflow jobs.

21.3 Triggers

The rule of thumb is not to use a field in the workflow itself which was used in the trigger condition.

22 Business Rules

The usage of Business Rules versus custom JavaScript logic is an important part of the architectural decisions. This chapter is therefore only relevant if the usage of Business Rules is at all applicable.

> ***Important Note***
> Business rules do create JavaScript, so coordinating any creation of business rules with any creation of extensions using custom JavaScript is critical.

22.1 Naming

Name Business Rules consistently according to their purpose:

Area	Purpose	Naming Schema
Visibility	Only visibility is set for one or several fields	Set Visibility of <Field Names>
Required Level	Only required level (mandatory, optional) is set for one or several fields	Set Required Level for <Field Names>
Prevent Change	Prevention of field change according to certain conditions	Prevent Change of <Field Name>
Population	Population of certain fields	Populate/Set <Field Names> [Based On <Related Fields>]
Lock/Unlock	Lock or unlock certain fields	Lock|Unlock <Field Names> [On <Table>]
Advanced Logic	Different types of actions are performed on several fields	Handle <Name of Area, e.g. “Backdate Validation on Case”>

22.2 Business Rule Template

Use the following table:

Rule Name	Table	Type	Event	Business Logic/ Automation	Fields Affected by Action
<Name>	<Table Name (if not mentioned in chapter>	Synchronous Workflow | Business Rule	Create | Update [<Field Name>]	<Logic>	<Fields>

22.3 Structure and Refactoring

For rules tied to the same table, continuously look for optimizations to avoid having too many rules. For example, if two rules set the visibility of two fields according to the same condition, they shall be combined.

23 Solution Management

Solutions are the way to transport the development artefacts from one system to another.

Note
The setup of Solutions is simplified and supported by Avanade BizApps Core including the automated deployment to the different environments with preconfigured pipelines.

23.1 Solution Slicing

Important Note
This information was updated due to new inputs in December 2021.

It is important to define the right solution split. For more information on the solution slicing topic, see the following sources:

- DevOps Process Documentation (chapter 4.9 Solution Slicing)
- D365 CE Solution Lifecycle Management

When defining the solution split, you should consider the following main points:

- Importing solutions changes the database structure of the Dataverse organization and this requires time as well the publishing/promotion process. Smaller solutions are more portable, and the time needed for import is smaller. (For more guidance, for example on patching, see DevOps Process Documentation).
- Smaller Solutions improve quality because you only import what has changed.
- When defining the solution structure for a project, you should always have a base solution which contains component that are or might be commonly used by multiple solutions. Having such a base solution makes it easier to onboard other apps (potentially built by other project teams) to the same organization. Solutions sitting on top of the base solution should resemble a functional split (for example one solution for Case Management, another for Opportunity Management).
- In Avanade projects, unmanaged solutions were often used in the past for various reasons (see DevOps Process Documentation). Over the last years the recommendation has shifted to using managed solutions. Nevertheless, unmanaged solutions are still possible and maybe necessary in some cases (see TOD for a typical technical split when unmanaged solutions are used).

23.2 Standardization of Publisher and Solution Naming

Getting the naming of publishers and solutions right, can be quite tricky.

The following sections describe proven practices and guidelines from the Avanade DSS Asset team which are also applicable to projects (differences between assets and projects are highlighted).

Note
The guidelines are incorporated into the Avanade BizApps Core tooling so that when setting up publishers or solutions, the risk of human error is minimized.

23.2.1 Why Standardize?

- Avoid collision with consulting service projects
- [Assets] Identify AVA assets in heavily customized systems
- Identify layered components, dependencies in complex solutions
- Reduce risk of „human factor”
- Professional appearances
- Follow Microsoft standards when possible

23.2.2 Projects

23.2.2.1 Publisher

- Typically it is sufficient to use a single publisher for all solutions (unlike for assets)
- Display name is usually the client name.
- Unique name is the project name without the client name.
- The prefix should have 7 characters and must be unique to the publisher.
- The choice value prefix should be an arbitrary number.

Display name	Name	Description	Prefix	Choice value prefix
My Client	myproject	A meaningful description.	clieprj	90000

23.2.2.2 Solutions

- “[Client] [Project] [Area]“ should be in the friendly name.
- “[Client] [Project]“ should not be in the unique name (for unique name use a combination of prefix and display name without client and project name).

Example:

Display name	Name	Publisher	Description
My Client My Project Case Management	clieprj_CaseManagement	My Client (myproject)	A meaningful description.

23.2.3 Assets

23.2.3.1 Publisher

- Dedicated Publisher for each solution (Microsoft uses single publisher but this is not necessary yet).
- Display name is always "Avanade".
- Unique name is the friendly name without "Avanade".
- The prefix should have 7 characters and must be unique to the publisher.
- The choice value prefix should be an arbitrary number to avoid collisions with other customizations on common assets.

Example:

Display name	Name	Description	Prefix	Choice value prefix
Avanade	exampleasset	A meaningful description.	avaexam	81000

23.2.3.2 Solutions

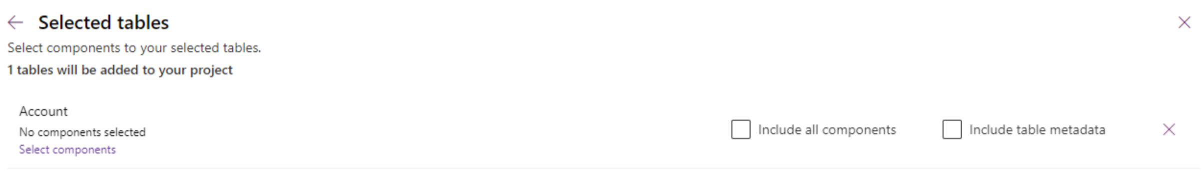
- "Avanade" should be in the friendly name.
- "Avanade" should not be in the unique name (for unique name use a combination of prefix and display name without "Avanade").

Example:

Display name	Name	Publisher	Description
Avanade Example Asset	avaexam_ExampleAsset	Avanade (exampleasset)	A meaningful description.

23.3 Add or Remove Solution Components

Its crucial to be aligned on the deployment process when you work with solutions. When you add a table to a solution you have two options “Include all components” or “Include table metadata”



23.4 Cleaning Up unused Solution Components

23.4.1 Fields

- Add “ZNotinUse” before all unused field display names
- Exclude “ZNotinUse” fields from Search (this can be done by editing several fields at once, from the field view in a solution)

It is then easier for users to only see relevant information when they use the Advanced Find and customize the views columns.

23.4.2 Forms

- Create new forms instead of using the default forms which will avoid the risk that system updates and ISV installations interfere with your forms.
- Add "ZZ " or “ZNotinUse” before all unused forms.
- If you create only one new form, use the table name as the form name for a better user experience.
- Remove all associated security roles on “ZNotinUse” forms and remove them from the User Apps.
- Uncheck the form for fall back.
- Deactivate the not in use forms.

As form status is not carried through with solution deployment, it is then easier for system customizers to deploy the solution and know on the target environment which form must be deactivated.

Alternatively, you can use the Deployment Tool (Avanade.DPA.Deployment) to deactivate forms or views in the target environment in a streamlined and consistent way.

23.4.3 Views

- Add "ZZ " or “ZNotinUse” before all unused views.
- Deactivate them.

As view status is not carried through with solution deployment, it is easier for system customizers to deploy the solution and know on the target environment which view must be deactivated.

Alternatively, you can use the Deployment Tool (Avanade.DPA.Deployment) to deactivate forms or views in the target environment in a streamlined and consistent way.

23.4.4 Dashboards

- Add all unused Dashboards to your project solution.
- Prefix them with ZZ.
- Deactivate them with security roles.

These customizations will follow solution deployments.

23.4.5 Release Information Task

As soon as there are unused fields, forms, views, dashboards, etc. which should be removed from the Organizations, create a Task in the Backlog, and add the specific items in the description of the work item.
Differentiate between steps to do before and steps to do after the solution import.
This makes it much easier to track the changes which must be done PRE or POST a deployment and can be used for the documentation.

23.5 Solution History

The solution history gives you visibility to export import tasks which are either currently running or have run.

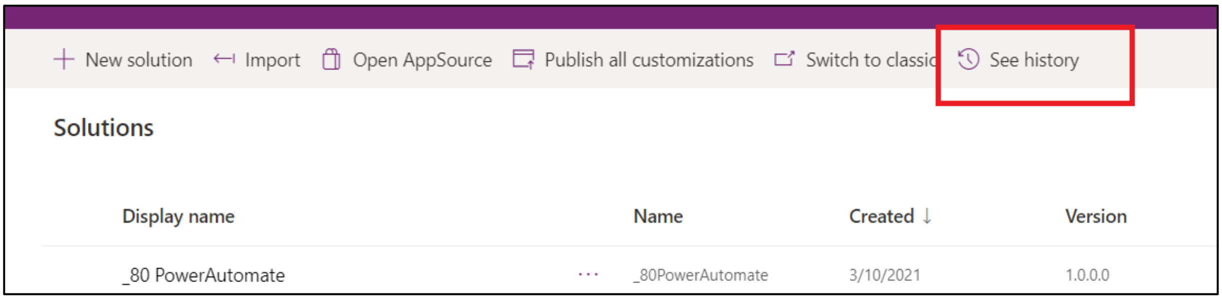


Figure 36 Access to solution history

If an operation is still running you can see if the End time column is still empty. The type of operation you can see in the operation column. You will see Microsoft imports and updates and your own.

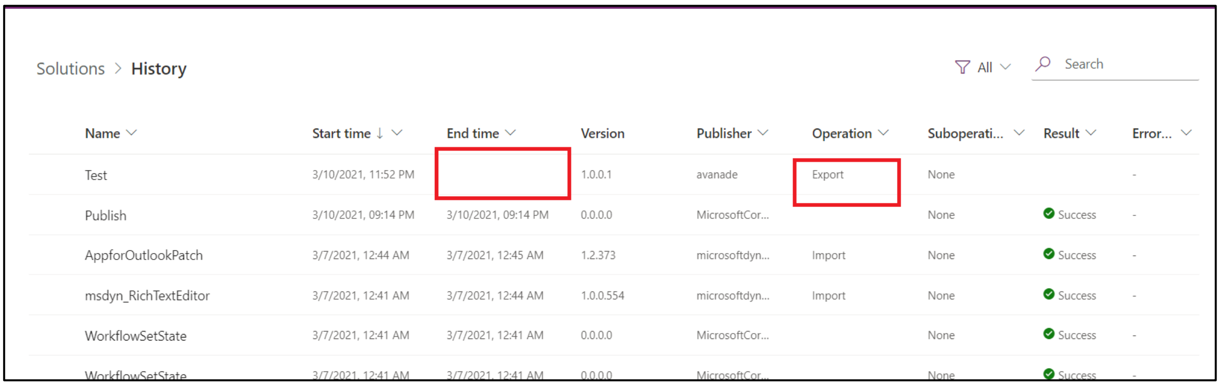


Figure 37 Solution history overview

When you click on the row, you get some more details.

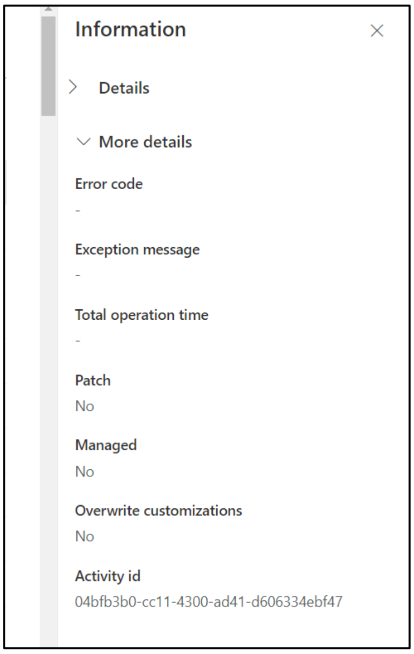


Figure 38 Solution History Details

23.6 Other Considerations

- Define the publisher as by the guidelines provided in 23.2.
- Always insert a meaningful description.
- Leave the Version Number as 1.0.0.0.
- Use Solution Checker to verify the quality of your solutions.

24 Changing Data Types

It is common that importing a solution that has the same schema name, but two different types, will fail to import.

There are several considerations when changing a schema name that already exists in the target environment, like Production.

Option 1

It is often easier to:

1. Create a new field
2. Set the correct display label on the new field
3. Change the display label for the old field to “<columnName> (not in use)” so that it is clear the field should not be used by end users.
4. Remove the old field from all forms, views, mapping, reports, etc.,
5. Disable the search feature of the now obsolete field.

This will prevent users from using it, while preventing any data loss.

Option 2

It is possible to migrate data from the old field into the new field:

1. Import a solution with the new field
2. Perform a migration of the data to the new field
3. Delete/hide the old field

25 Translation

Normally the translation uses the standard excel translation file. To translate you must select a solution and export the translation file.

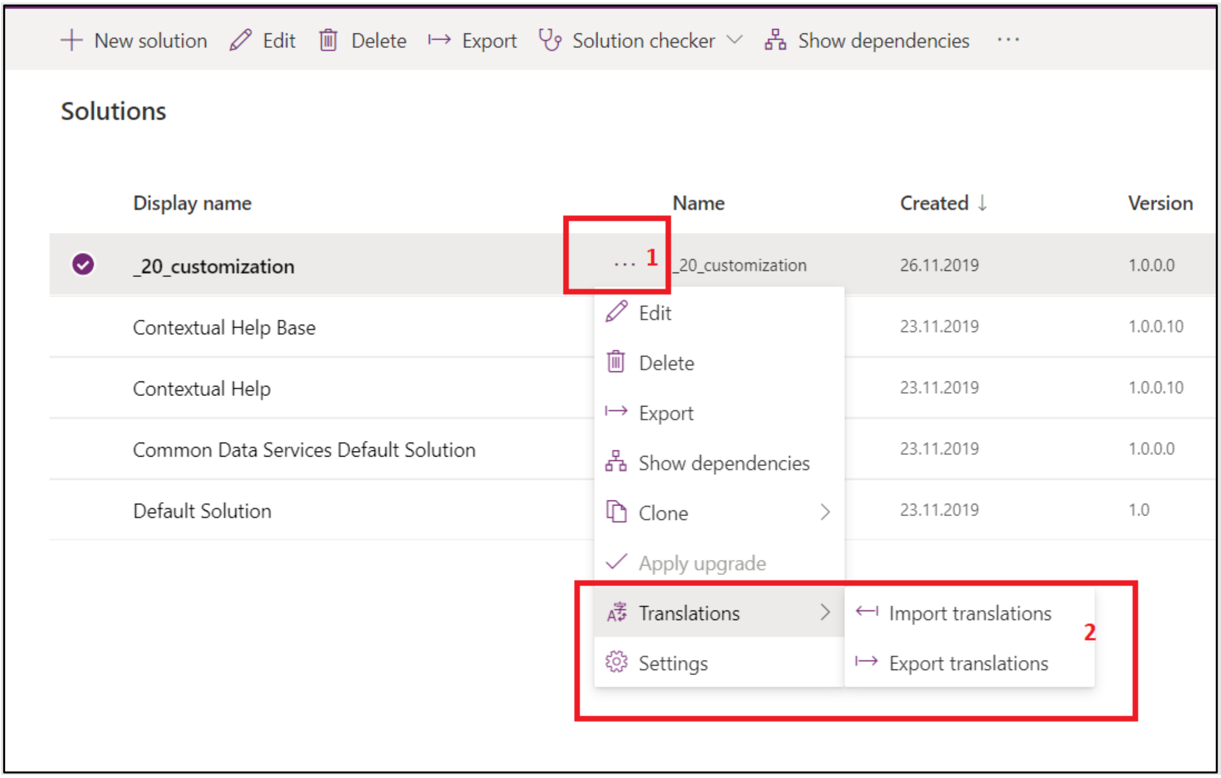


Figure 39 Translation file export and import

Recommendation

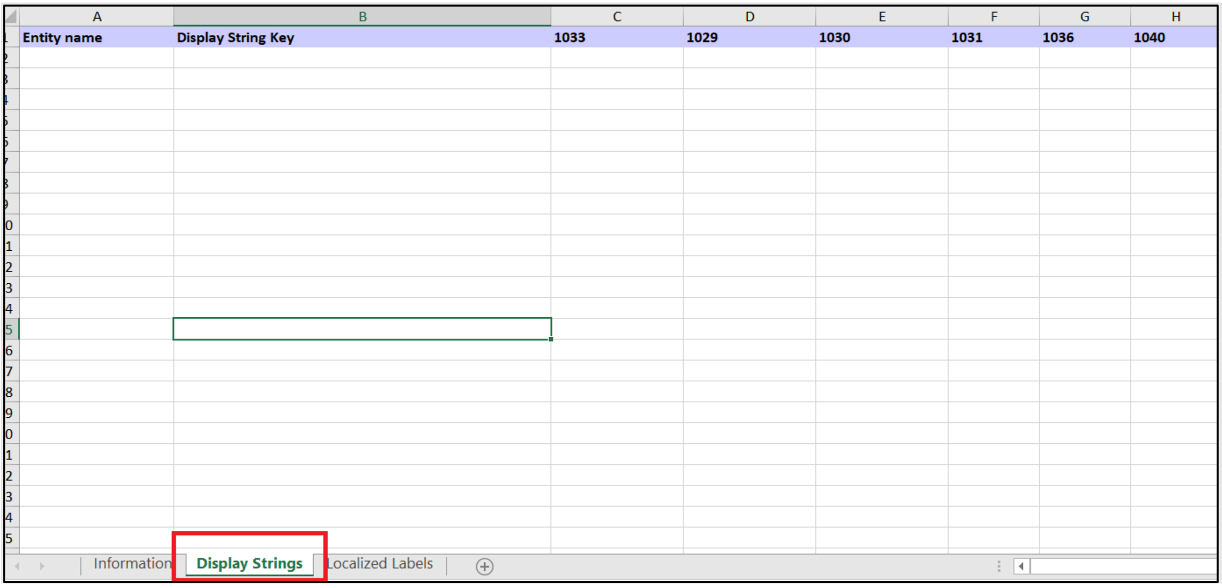
The translation files will only include the entities which are present in the selected solution. Use this behavior to exclude not needed translations for specific tables. E.g. create a translations solution which includes only the to be translated tables.

Important Note

The translation export will include everything from the table! If you only partially include the table into your solution you will still find ALL views, columns, and forms in the exported translation file.

- The translation zip contains two files:
  - CrmTranslation.xml and [Content_Types].xml
  - The translations are included in the CrmTranslation.xml file.
- Do your translations and afterwards upload the same file to this specific environment.
- Translation files are environment specific! The translation file will contain all labels from the tables and columns which were part of this solution. Additionally, the Buttons are part of the translation file.
- All button translations are distinguishable by the value “RibbonCustomization” in the first column.
- The buttons for each table are added after the form translations of the corresponding table.

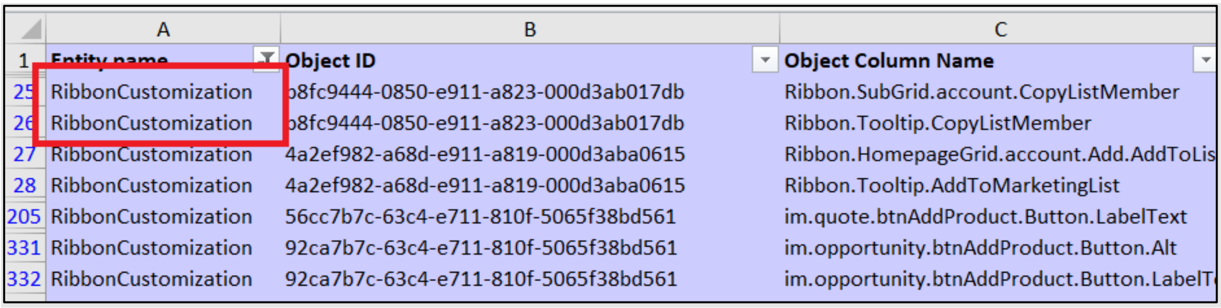
- Upload only changes. Remove all columns and/or rows which are not needed!
- Remove empty rows.
- The table “Display Strings” contains messages which are created by Microsoft and are rarely retranslated. Remove all rows that are not needed.



Entity name	Display String Key	1033	1029	1030	1031	1036	1040

Figure 40 Table Display Strings

If it happens that during the translation file import an error occurs.
The recommendation is to split the excel translation file and upload the ribbon customizations and the label customizations. In almost all cases this resolves the errors which occur.
To do so filter the “Table name” column first for contains “RibbonCustomization” and remove all these rows. Save the file.



	Entity name	Object ID	Object Column Name
25	RibbonCustomization	8fc9444-0850-e911-a823-000d3ab017db	Ribbon.SubGrid.account.CopyListMember
26	RibbonCustomization	8fc9444-0850-e911-a823-000d3ab017db	Ribbon.Tooltip.CopyListMember
27	RibbonCustomization	4a2ef982-a68d-e911-a819-000d3aba0615	Ribbon.HomepageGrid.account.Add.AddToLis
28	RibbonCustomization	4a2ef982-a68d-e911-a819-000d3aba0615	Ribbon.Tooltip.AddToMarketingList
205	RibbonCustomization	56cc7b7c-63c4-e711-810f-5065f38bd561	im.quote.btnAddProduct.Button.LabelText
331	RibbonCustomization	92ca7b7c-63c4-e711-810f-5065f38bd561	im.opportunity.btnAddProduct.Button.Alt
332	RibbonCustomization	92ca7b7c-63c4-e711-810f-5065f38bd561	im.opportunity.btnAddProduct.Button.LabelT

Figure 41 Filtered by RibbonCustomization

Afterwards revoke your changes and filter on “Table name” where not contains “RibbonCustomization” also save this file.
Use these files for the upload.

> ***Note***
> For further technical reference regarding translations, see https://docs.bizapps.avanade.com/3.0/getting-started/next-steps/translations/.

25.1 App Name translation

The App name can be translated and is part of the translation excel if the sitemap is part of the solution.

Entity name	Object ID	Object Column Nam	1033
Solution	5b318104-cf81-eb11-a812-0022481d335d	friendlyname	Test
Publisher	9312dcf2-ce81-eb11-a812-0022481d335d	friendlyname	Avanade
SiteMap	a6a0109f-e281-eb11-a812-0022481d335d	sitemapname	Test App

Figure 42 App Name Translation

You will find the App Name if you filter the first column on the value “SiteMap”.

25.2 Sitemap translation

The Navigation (Sitemap) of an App is not part of the “normal” translation file in Dynamics 365.
If you changed and/or added additional area, group, main or sub navigation items you must take care to directly translate them.
The recommended approach would be to use the internal Sitemap Designer to translate these navigation points.
To do this for the Classic Site Map

1. Add the Site Map to your solution
2. Double click on the Site Map
3. The Sitemap Designer opens

4. Highlight the sub navigation (1) on the left (2) via Advanced >> Titles
5. You can add the required languages using Advanced, Titles.
6. Existing languages can be edited by directly clicking on the label in the list.

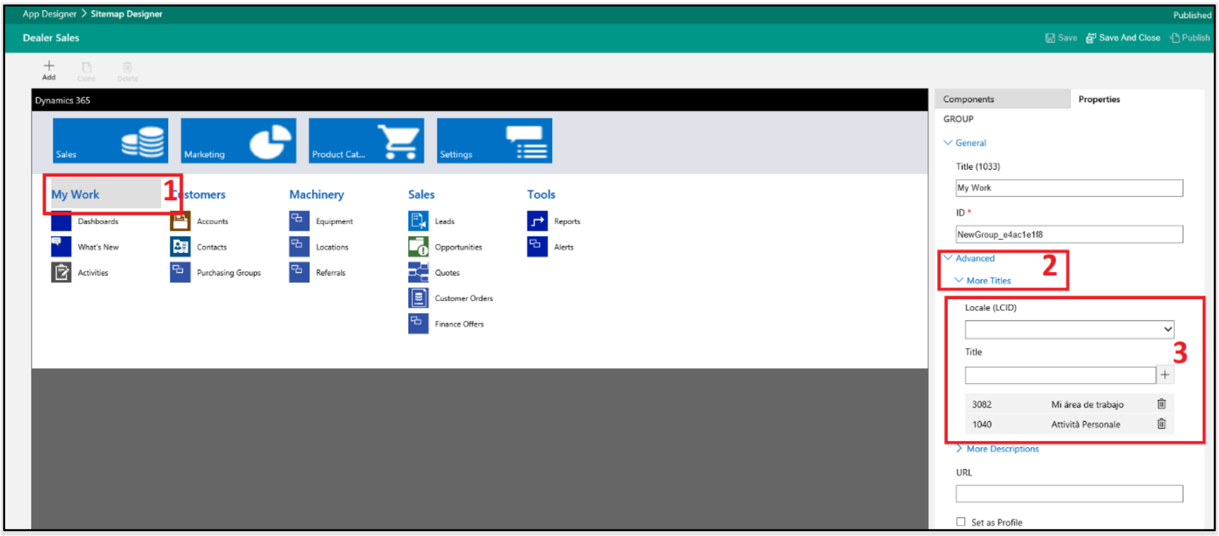


Figure 43 Sitemap Designer Translation

You must translate the sitemap in your customization system!
The reason behind this is, that the sitemap is part of the app you customize and is overwritten during a solution import.

Important Note
To be able to translate the sitemap you do not need to activate any additional language in the environment.

Note
If for any reason the sitemap designer does not work for you then export the specific sitemap only in a temporary solution and add the translation labels in the xml structure of the sitemap. This is the supported approach by Microsoft for changing translation of Site Map Labels.

25.3 Field Translations

Dynamics 365 supports translations of content e.g. Product Names.
This must be handled carefully and currently there is no automated tooling approach available in the DSS Framework.

You can reach the translations via “Settings” → “Data Management” → “Export Field Translations” / “Import Field Translations”.

26 Business Units

The root Business Unit is by default named the same way as the Organization itself. After a copy process of an Org via Power Admin Portal, the name is just getting changed to the technical name (e.g. org49bbfc81).

It is best practice to streamline the Root-BU renaming in all environments to avoid issues between the stages (Dev/Test/Prod).

Since the renaming is not possible via the UI (the Parent-Lookup field is mandatory), Avanade prepared different options to update the BU via the internal tooling.

Option 1 – Use Entity Management Cockpit (EMC)
EMC can be used to run it via UI or Command line and update the BU to the required name.
A complete guide can be found in the DSS Framework Wiki: Scenario 2 - Set Business Unit Structure - Overview (visualstudio.com). This guide also shows how to roll out complete Business Unit hierarchies.

Option 2 – Use Deployment Tool and add it to your Pipeline
To avoid manual overhead, the recommendation is to just streamline the renaming as part of the deployment pipeline. A complete configuration (XML) is available in the Deployment Tool Samples: UpdateRootBusinessUnit.xml - Repos (visualstudio.com).

27 Configuration Questions

27.1 Enable Quick Find Limit

- In System Settings, leave the "Enable Quick Find Limit" to Yes or set it to Yes if it is not set to Yes.
- This has a huge impact on the time it takes to return results from a Quick Find query.
- Try to limit the number of Search Columns (especially lookup fields) and filter the Quick Find view to exclude as many rows as possible.

27.2 Performance Checklist

Ideally your initial architecture should include architecting for scalability and performance.

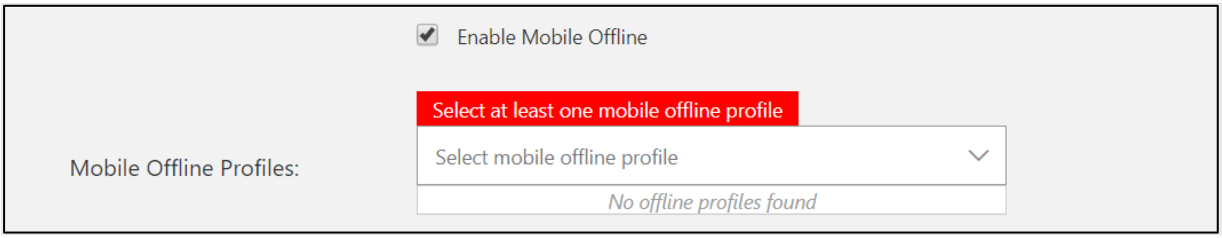
Additionally check anything which is relevant in the Performance Checklist in order to optimize the performance of your solutions and apps.

Key areas that impact performance include

- Business Unit Design
- Security Role Design
- Amount of records in tables
- JavaScript management and design
- Form Layout
- Flow Variables

28 Mobile Offline configuration

Microsoft has an out of the box solution to support offline capabilities. This is done via the mobile application. To be able to use this you must customize the app to enable mobile offline.



There is currently no good way to transfer this property.
The offline profiles are solution aware but after you include them in a solution there is a known Bug from MSFT open that specific other customization types are added to the solution after you choose to download it.
Consider this especially if you need to export managed solutions.

29 Hints/General Knowledge/Tools

29.1 Tools: XrmToolbox

The XrmToolbox provides a very good list of plugins which will help you in your daily customization life. Its available here: https://www.xrmtoolbox.com

Recommendation
Access to the customer environments via XrmToolbox should be possible! Otherwise there is potential of either additional work for customization changes. Up to even not possible.

29.2 Do not Update Users or Business Units Too Frequently

An update on a Business Unit or User requires elevated privileges and may also invalidate the security cache. If you need frequent updates on Business Unit, consider creating a dedicated custom table for better system performance.

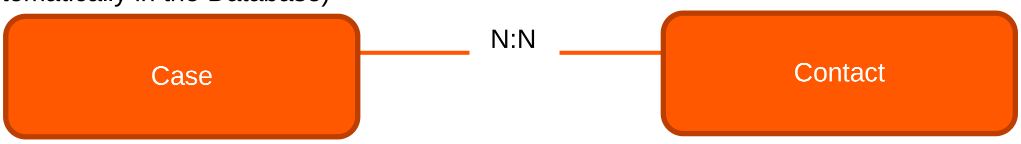
29.3 Verify Customizing Guidelines

If you mistyped a column or table name and saved, recreate it! But not by all means. If data is already existent and/or the sprint is nearing its end you must decide if it is worth the risk!

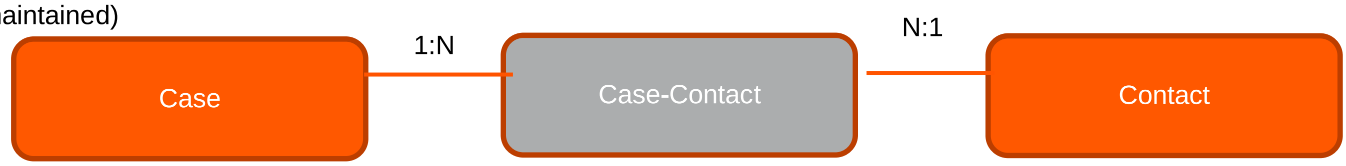
29.4 Intersect Tables / Custom N:M relations

In case there is a need for custom N:M relations (e.g. Activity Assignment, Attachment Assignment), the goal is to hide this additional table as good as possible from the end user. Otherwise this will end in endless discussions with the customer. Following patterns must be followed if there is no explicit reason for not doing it (e.g. dedicated requirement to not go for a Sub grids).

Out of the Box – Many to Many relation (no Intersect to be maintained, everything happens automatically in the Database)

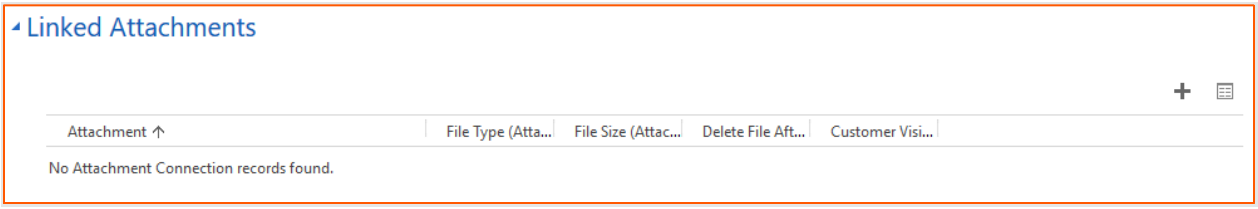


Custom – Many to Many relation (Intersect “Case-Contact” to be created and maintained)



29.4.1 Sub grids for Intersect Tables

Since the users are not interested in the data model, they should not be forced to jump to “Related Tables” section inside of the form. The behavior should be straight forward. A Sub grid is required for every Intersect Table.



29.4.2 Quick Create for Intersect Tables

To avoid the standard behavior of selecting existing row or clicking the + icon in the lookup dialog, a Quick Create must be available and the lookup in the Intersect rows needs to be required to ensure that clicking the + icon in the Sub grid directly opens the Quick Create Form. The Quick Create Form should only contain the relevant fields.

In this sample, the user is not forced to add anything except the Attachment itself. So, the Intersect Table is basically hidden for the user.

29.4.3 Set Name Field for the Intersect Table

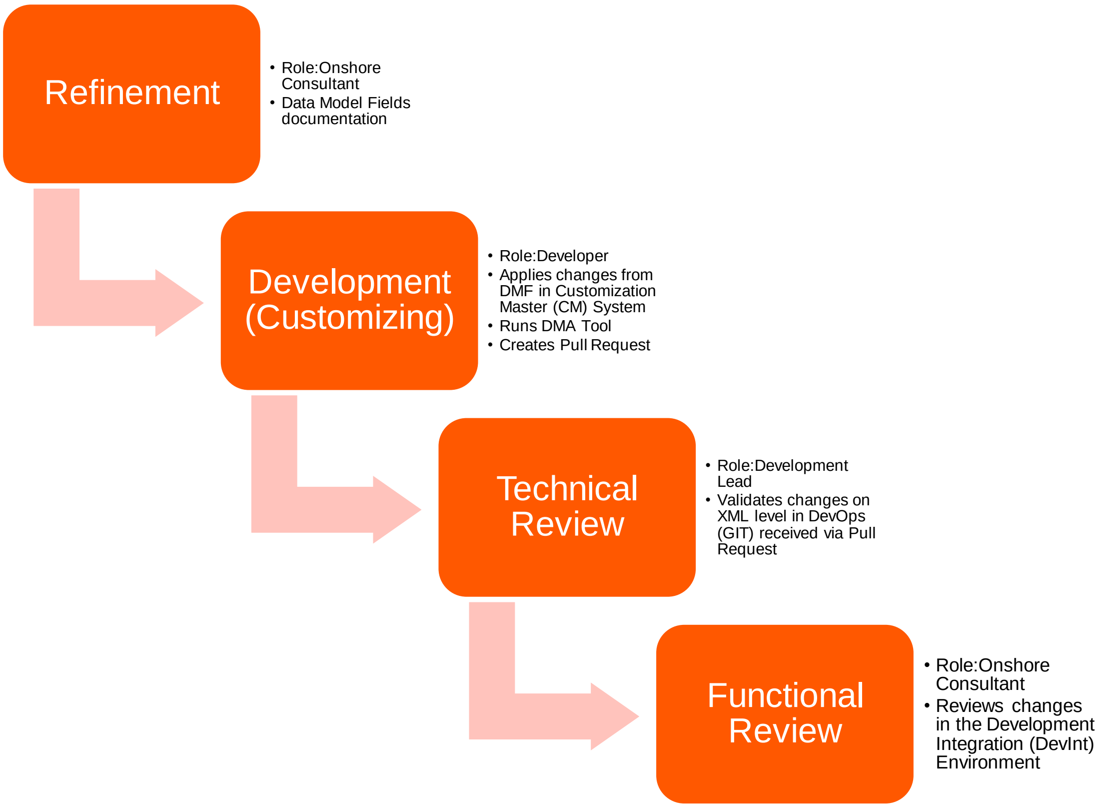
By default, the Intersect Table Name (primary table field) must be set automatically by business Rule (or Workflow if more advanced logic is required). Only if explicitly requested by the customer, we leave it up do the user to define the name. In most of the cases, a static value like “Attachment Connection” + <current date> is sufficient.

30 Customization Processes

Every project might define its own process for customizing depending on the overall setup (onshore/nearshore/offshore/customizing done by developers or consultants…).
This chapter lists an overview of patterns used in existing projects.

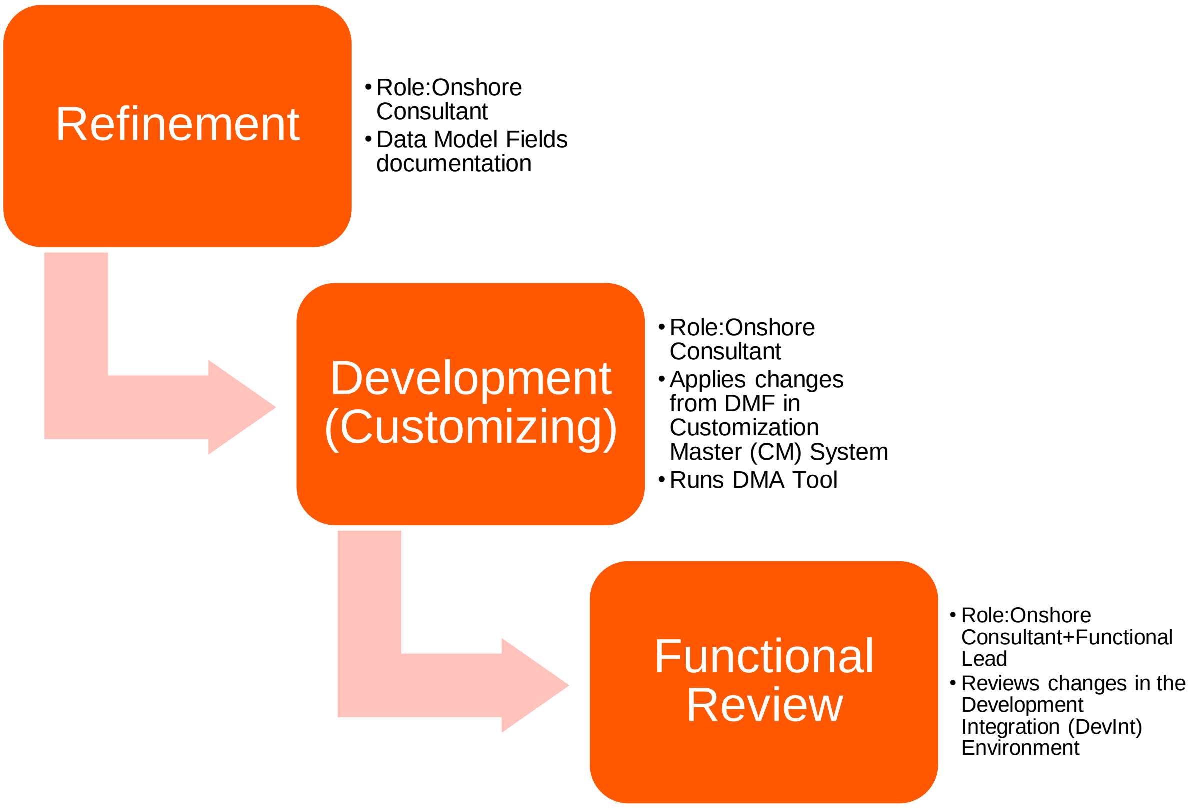
30.1 Approach A – Onshore/Offshore split using Pull Requests

The following setup was used in an existing enterprise project.



30.2 Approach B – Onshore without Pull Request

The following setup was used in an existing enterprise project.



31 Appendix

31.1 Technical Solution Split

Note
This is only relevant when using an unmanaged approach.

Following content should only be put in the corresponding solutions. Don't mix the solution content!

Solution name	Content of the solution
AvanadeDSSFrameworkCoreSolution	Global configuration solution
<solutionprefix>_10WebResources	All web resources – JavaScript, HTML, Images, XML
<solutionprefix>_20Customizing	All customizations including tables, forms, and views.
<solutionprefix>_30SecurityRoles	All security roles
<solutionprefix>_40Plugins	All Plugins
<solutionprefix>_50Workflows	All Workflow and Process dialogs.
<solutionprefix>_60Flows	All Flows
<solutionprefix>_70Templates	All Templates (E-Mail, Word mail merge etc.)
<solutionprefix>_80Reports	All Reports

Exceptions:

- **Classic Business Process Flows** are technically processes. Don't put them in the process solution, because starting from Dynamics 365 version 8, each Business Process Flow also creates a table. You'll need this table in the instance before importing the solution and the table needs the business process flow.
- Put business process flows in the customization solution.
- **Security Profiles** are security content but require customizations for the columns before you can import them. And the customization requires the Security Profile to be there otherwise it will not import. → Therefore, put them in the customization solution.

31.2 Document Notes

Throughout the document Avanade will summarize the outputs of each section. These outputs can be categorized under the following headings:

Note
This header is to make the reader aware of something specific in the document and will give some additional context to the section.

Important Note
This header is to ensure the reader is fully aware of the point being highlighted. The information provided should be fully considered when understanding the context of the section.

Recommendation
A recommendation being made by Avanade, but not necessarily a design decision.

Assumption

Based on the workshops and knowledge of Avanade assumptions on configurations and requirements are listed.

Design Decision

A design decision based on Avanade/Microsoft recommended best practices.