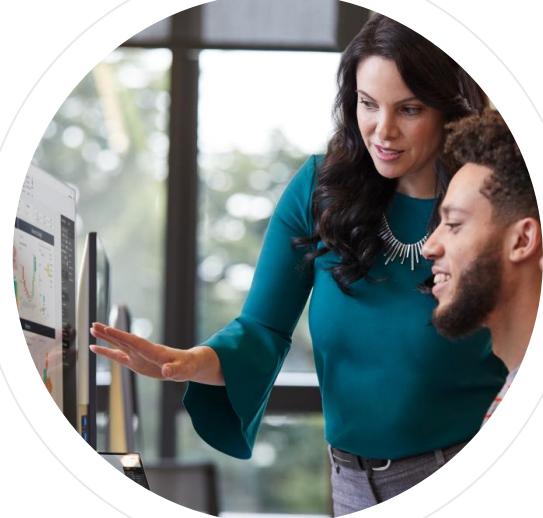




PL-400T00

Microsoft Power Platform

Developer



© Copyright Microsoft Corporation. All rights reserved.

1

Course description



The Microsoft Power Platform helps organizations optimize their operations by simplifying, automating and transforming business tasks and processes. In this course, students will learn how to build Power Apps, Power Automate flows and extend the platform to complete business requirements and solve complex business problems.



The course material includes a combination of instructor-led presentation and discussion along with hands-on lab activities

© Copyright Microsoft Corporation. All rights reserved.

2

Course Agenda

- LP01: Work with Microsoft Dataverse
- LP02: Create model-driven apps
- LP03: Create canvas apps
- LP04: Advanced techniques in canvas apps
- LP05: Automate a business process using Power Automate
- LP06: Introduction to developing with Power Platform
- LP07: Extending the model-driven apps user experience

© Copyright Microsoft Corporation. All rights reserved.

3

Course Agenda

- LP08: Create code components with the Power Apps Component Framework
- LP09: Extending Microsoft Dataverse
- LP10: Integrate Dataverse and Azure
- LP11: Custom Connectors
- LP12: Application Lifecycle Management (ALM)
- LP13: Create a Technical Design

© Copyright Microsoft Corporation. All rights reserved.

4

Certification areas (PL-400)

Study Areas	Weights
Create a technical design	10-15%
Configure Microsoft Dataverse	15-20%
Create and configure Power Apps	10-15%
Extend the user experience	10-15%
Extend the platform	35-40%
Develop Integrations	5-10%

This course is aligned with the exam PL-400 Microsoft Power Platform Developer

Percentages indicate the relative weight of each area on the exam

The higher the percentage, the more questions you are likely to see in that area

© Copyright Microsoft Corporation. All rights reserved.

5

Prerequisites



Candidates should have development experience that includes Microsoft Power Platform services, JavaScript, JSON, TypeScript, C#, HTML, .NET, RESTful Web APIs, and Microsoft Azure.

© Copyright Microsoft Corporation. All rights reserved.

6

Preparing for the labs

https://microsoftlearning.github.io/PL-400_Microsoft-Power-Platform-Developer/

You will work in a virtual machine environment to complete the labs. The virtual machines used for this course are hosted by an authorized lab hosting provider.

The detailed lab instructions will be included in your lab environment.



© Copyright Microsoft Corporation. All rights reserved.

7

WARNING – You may experience UI lab discrepancies

- Given the dynamic nature of Microsoft cloud tools, you may experience Microsoft Power Platform user interface (UI) changes that were made following courseware development that do not match up with lab instructions.
- While Microsoft Learning will make every attempt to keep lab instructions current with the latest UI changes, you may run into Microsoft Power Platform changes that occur before Microsoft Learning becomes aware of them.
- If this occurs, you will have to adapt to the UI changes and work through them in the labs, as necessary.



© Copyright Microsoft Corporation. All rights reserved.

8

Access to labs

Once you have completed this course, you can continue to access its labs within GitHub for personal self-study.

The GitHub link can be found on the Course Introduction page of your student manual.



© Copyright Microsoft Corporation. All rights reserved.

9

Microsoft Power Platform moves fast

- Updates happen often, sometimes weekly!
- Course content is updated regularly, but you will often find something is a little different here and there.
- Don't panic, release notes are public.
- <https://learn.microsoft.com/power-platform/release-plan/2023wave2/>

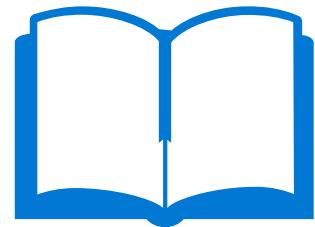
© Copyright Microsoft Corporation. All rights reserved.

10

Student materials on Microsoft Learn

PL-400 Learn Collection

<https://learn.microsoft.com/en-us/users/sarikulthm-2667/collections/12omu31mm02ond>



© Copyright Microsoft Corporation. All rights reserved.

11

Learning Path 0 practice labs

Lab 0: Validate lab environment



© Copyright Microsoft Corporation. All rights reserved.

21



© Copyright Microsoft Corporation. All rights reserved.

22



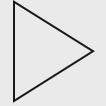
Learning Path 1: Work with Microsoft Dataverse

© Copyright Microsoft Corporation. All rights reserved.



23

Module 1: Introduction to Microsoft Dataverse



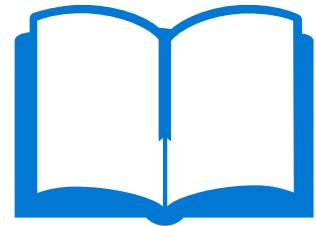
© Copyright Microsoft Corporation. All rights reserved.

24

Microsoft Learn module

Identify foundational components of Microsoft Power Platform

<https://learn.microsoft.com/training/modules/introduction-common-data-service>

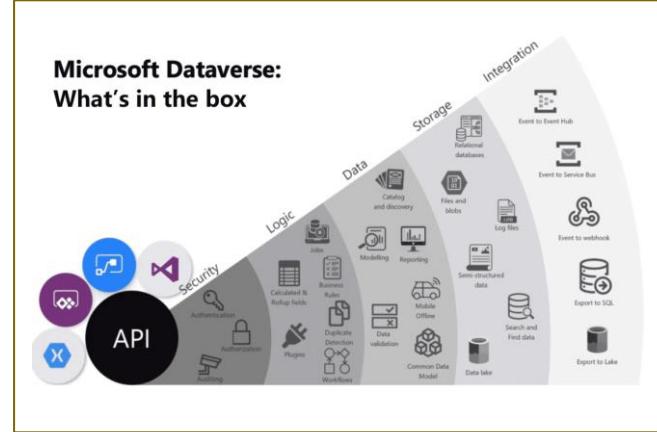


© Copyright Microsoft Corporation. All rights reserved.

25

Dataverse

Microsoft Dataverse is a cloud-based, low-code data service and app platform, that easily structures a variety of data and business logic to support interconnected applications and processes in a secure and compliant manner



© Copyright Microsoft Corporation. All rights reserved.

26

Benefits of Microsoft Dataverse

- **Simple to manage** - Both the metadata and data are stored in the cloud. You don't need to worry about the details of how they're stored.
- **Helps to secure data** - Data is stored so that users can see it only if you grant them access. Role-based security allows you to control access to tables for different users within your organization.
- **Access your Dynamics 365 data** - Data from your Dynamics 365 applications is also stored within Dataverse, which allows you to quickly build apps that use your Dynamics 365 data and extend your apps by using Power Apps.
- **Rich metadata** - Data types and relationships are used directly within Power Apps.
- **Logic and validation** - Define calculated columns, business rules, workflows, and business process flows to ensure data quality and drive business processes.
- **Productivity tools** - Tables are available within the add-ins for Microsoft Excel to increase productivity and ensure data accessibility.

© Copyright Microsoft Corporation. All rights reserved.

27

Microsoft Dataverse defined



A Dataverse data store is a single instance of Microsoft Dataverse which stores data in a set of standard and custom data structures called tables.



A Dataverse data store is scalable and supports both large data sets and complex data models.



The standard table design in a Microsoft Dataverse data store is based upon an open data model standard called Common Data Model. The structure of a Microsoft Dataverse data store is based upon the definitions and schema in the Common Data Model.

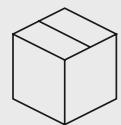


Dataverse contains comprehensive business logic, processing, integration, and security features that aid the building of business solutions.

© Copyright Microsoft Corporation. All rights reserved.

28

Module 2: Manage environments



© Copyright Microsoft Corporation. All rights reserved.

29

Microsoft Learn module

Create and manage environments in Dataverse

<https://learn.microsoft.com/training/modules/create-manage-environments/>

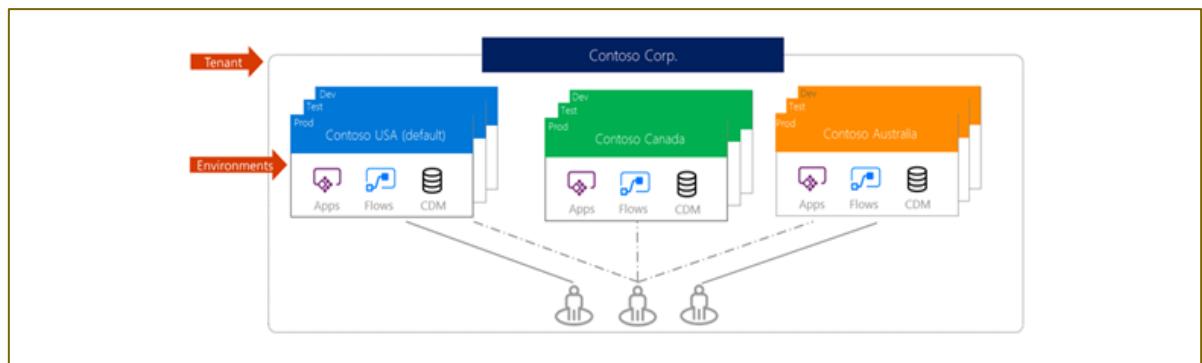


© Copyright Microsoft Corporation. All rights reserved.

30

Environments in Power Platform

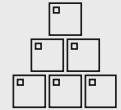
Power Platform environments are used to store, manage, and share your organization's business data, apps, flows, and users.



© Copyright Microsoft Corporation. All rights reserved.

31

Module 3: Manage customizations with solutions



© Copyright Microsoft Corporation. All rights reserved.

32

Solutions

- Solutions are containers of Power Platform components
- Transport customizations from one environment to another
- Part of an overall strategy for application lifecycle management (ALM)



© Copyright Microsoft Corporation. All rights reserved.

33

Create solution

Display name used in Maker portal

Name is the schema name and cannot be changed and cannot have spaces

Publisher defines the prefix for new components

Version number to track updates

New solution

Display name *

Name *

Publisher *

Select a Publisher

+ New publisher

Version *

1.0.0

More options

© Copyright Microsoft Corporation. All rights reserved.

34

Components in solutions

- Create new components in solutions
- Add existing components to solutions

Power Apps

Search

+ New

Add existing

PL-400 Demos > All

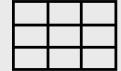
Name	Type	Managed
(empty)		

We didn't find anything to show here

© Copyright Microsoft Corporation. All rights reserved.

35

Module 4: Create and manage tables in Microsoft Dataverse



© Copyright Microsoft Corporation. All rights reserved.

36

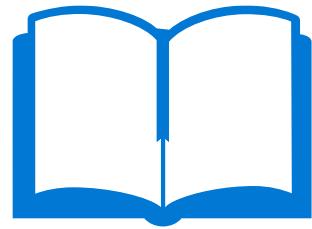
Microsoft Learn module

Create tables in Dataverse

<https://learn.microsoft.com/training/modules/get-started-with-powerapps-common-data-service>

Manage tables in Dataverse

<https://learn.microsoft.com/training/modules/create-manage-entities>



© Copyright Microsoft Corporation. All rights reserved.

37

Identify tables in Microsoft Dataverse

A table is a logical structure containing rows and columns that represents a set of data.

Full Name	Email	Company Name	Business Phone
Jim Glynn (sample)	someone_j@example.com	Coho Winery (sample)	555-0109
Maria Campbell (sample)	someone_d@example.com	Fabrikam, Inc. (sample)	555-0103
Nancy Anderson (sample)	someone_c@example.com	Adventure Works (sample)	555-0102
Patrick Sands (sample)	someone_k@example.com	Alpine Ski House (sample)	555-0110
Paul Cannon (sample)	someone_h@example.com	Alpine Ski House (sample)	555-0107

© Copyright Microsoft Corporation. All rights reserved.

38

Logic and validation

Tables within Dataverse can leverage rich server-side logic and validation to ensure data quality.



Business rules: Business rules validate data across multiple columns in a table, and provide warning and error messages, regardless of the app that's used to create the data



Business process flows: Business process flows guide users to ensure they enter data consistently and follow the same steps every time. Business process flows are currently supported only for model-driven apps



Workflows: Dataverse classic workflows automate business processes without requiring user interaction



Business logic with code: Business logic supports advanced developer scenarios that extend the application directly through code

© Copyright Microsoft Corporation. All rights reserved.

39

Types of tables

The three types of tables you can create are:



Standard

Create and edit records



Activity

Used to track interactions
on standard tables



Virtual

Connect to tables in an
external data source

© Copyright Microsoft Corporation. All rights reserved.

40

Record ownership

When you create a custom table, the options for ownership are:



User or team owned: Actions that can be performed on these rows can be controlled at the user level.



Organization-owned: Access to the data is controlled at the organization level.

© Copyright Microsoft Corporation. All rights reserved.

41

Table metadata

Microsoft Dataverse is designed to let you quickly and easily create a data model for your application, based on the tables and the table metadata that you include in your app.

- Display name
- Plural name
- Schema name
- Primary column
- Table type
- Record ownership
- Advanced options

The screenshot shows the 'New table' configuration page. It includes fields for 'Display name', 'Plural name', 'Description', and 'Schema name'. Under 'Advanced options', there are fields for 'Type' (set to 'Standard') and 'Record ownership' (set to 'User or team'). A checkbox for 'Enable attachments (including notes and files)' is present. The 'Primary column' field is highlighted with a red border.

© Copyright Microsoft Corporation. All rights reserved.

42

Advanced table options

- Enable attachments
- Create activities
- Auditing
- Duplicate detection
- Connections
- Quick create
- Offline

Note: some options cannot be turned off after they have been enabled

The screenshot shows advanced configuration options for a table. It includes sections for 'Choose table image' (with a dropdown menu and a 'New image web resource' button), 'Color' (with a color picker), and 'For this table' (checkboxes for 'Apply duplicate detection rules', 'Track changes', 'Provide custom help', 'Audit changes to its data', 'Leverage quick-create form if available', and a 'Help URL' input field). Below these are sections for 'Make this table an option when' (checkboxes for 'Creating a new activity', 'Doing a mail merge', and 'Setting up SharePoint document management'), 'Rows in this table' (checkboxes for various row-level properties like 'Can have connections', 'Can appear in search results', etc.), and a note about enabling certain options.

© Copyright Microsoft Corporation. All rights reserved.

43

Module 5: Create and manage columns in Microsoft Dataverse



© Copyright Microsoft Corporation. All rights reserved.

44

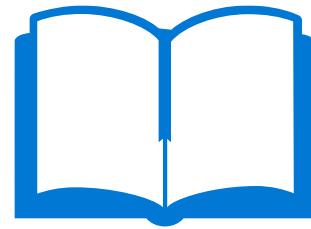
Microsoft Learn module

Create and manage columns within a table in Dataverse

<https://learn.microsoft.com/training/modules/create-manage-fields-within-entity>

Working with choices in Dataverse

<https://learn.microsoft.com/training/modules/working-with-option-sets>



© Copyright Microsoft Corporation. All rights reserved.

45

Columns in Microsoft Dataverse

Columns are a way to store a discrete piece of information within a row in a table.

Display name	Name	Data type	Managed	Customizable	Required	Searchable
(Deprecated) Process Stage	StageId	Ed Unique identifier	Yes	No	No	No
(Deprecated) Traversed Path	TraversedPath	All Single line of text	Yes	Yes	No	No
Account	AccountId	Ed Unique identifier	Yes	Yes	Yes	No
Account Name	Name	All Single line of text	Yes	Yes	Yes	Yes
Account Number	AccountNumber	All Single line of text	Yes	Yes	No	Yes
Account Rating	AccountRatingCode	Ed Choice	Yes	Yes	No	Yes
Address 1	Address_1Composite	Multiple lines of text	Yes	Yes	No	Yes
Address 1: Address Type	Address1_AddressTypeCode	Ed Choice	Yes	Yes	No	Yes
Address 1: City	Address1_City	All Single line of text	Yes	Yes	No	No
Address 1: Country/Region	Address1_Country	All Single line of text	Yes	Yes	No	Yes
Address 1: County	Address1_County	All Single line of text	Yes	Yes	No	Yes
Address 1: Fax	Address1_Fax	All Single line of text	Yes	Yes	No	No
Address 1: Freight Terms	Address1_FreightTermsCode	Ed Choice	Yes	Yes	No	Yes
Address 1: ID	Address1_AddressId	Ed Unique identifier	Yes	No	No	No
Address 1: Latitude	Address1_Latitude	Ed Real	Yes	Yes	No	No

© Copyright Microsoft Corporation. All rights reserved.

46

Add a column to a table

You can add columns when you create a new custom table, or you can add columns to an existing table at any time.

- Display name
- Descriptions
- Data type
- Format
- Behavior
- Required
- Searchable
- Schema name
- Advanced options

New column
Previously called fields. [Learn more](#)

Display name *

Description

Data type *

Format *

Behavior *

Required *

Searchable *

Advanced options ^

Schema name *

dem_

© Copyright Microsoft Corporation. All rights reserved.

47

Column data types

	Single line of text		Currency		Choices
	Multiple lines of text		Date and time		Yes/No
	Whole number		Lookup		File
	Floating point number		Customer		Image
	Decimal number		Choice		

© Copyright Microsoft Corporation. All rights reserved.

48

Other column details

	Column display formats		Column security
	Column properties		Auditing
	Column requirement		Description
	Searchable		

© Copyright Microsoft Corporation. All rights reserved.

49

Date and Time data type

- Dates stored in UTC
- Date and Time or Date Only
- Time zone adjustment
 - User Local
 - Date only
 - Time zone independent

The screenshot shows the configuration for a 'Date and Time' data type. The fields are:

- Data type: Date and time
- Format: Date only
- Behavior: Simple
- Required: Optional
- Searchable: Checked
- Schema name: dem_
- Time zone adjustment: User local (selected)

A dropdown menu for 'Time zone adjustment' shows three options: Date only, User local (selected), and Time zone independent.

© Copyright Microsoft Corporation. All rights reserved.

50

Choice data type

Choices provide an exact list of values that people can choose.

- List of values
- Key/Value pairs
- Global vs Local

The 'User Information' section includes a 'Preferred Phone' field with a dropdown menu containing 'Main Phone', 'Other Phone', 'Home Phone', and 'Mobile Phone'.

The 'New choice' dialog shows a list of choices for 'PC Type' with the following data:

Label	Value
Laptop	930.620.000
Desktop	930.620.001
Tablet	930.620.002
Server	930.620.003

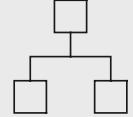
The 'Data type' configuration for 'Choice' includes:

- Data type: Choice
- Behavior: Simple
- Required: Optional
- Searchable: Checked
- Sync with global choice?:
 - Yes (recommended) - Can be used in multiple tables, and will stay updated everywhere.
 - No - Creates a local choice that can only be used in one table. People using it can add new choices.
- Sync this choice with: New choice
- Default choice: None

© Copyright Microsoft Corporation. All rights reserved.

51

Module 6: Create relationships between tables in Microsoft Dataverse



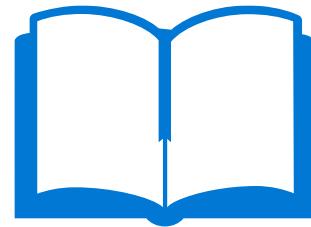
© Copyright Microsoft Corporation. All rights reserved.

52

Microsoft Learn module

Create a relationship between tables in Dataverse

<https://learn.microsoft.com/training/modules/create-relationship-between-cds-entities>



© Copyright Microsoft Corporation. All rights reserved.

53

Table relationships

Table relationships define the different ways table records can be associated with records from other tables or the same table.

- **One-to-many relationships:** In a one-to-many (1:N) table relationship, many related table records are associated with a single primary table record in a parent/child relationship.
- **Many-to-many relationships:** In a many-to-many (N:N) table relationship, many table records are associated with many other table records. Records that are related through N:N table relationships are considered peers.
- When I delete a record, should any records that are related to that record also be deleted?
- When I assign a record to a new owner, do I also have to assign all related records to the new owner?
- How can I streamline the data entry process when I create a new related record in the context of an existing record?
- How should people who view a record be able to view the related records?

© Copyright Microsoft Corporation. All rights reserved.

54

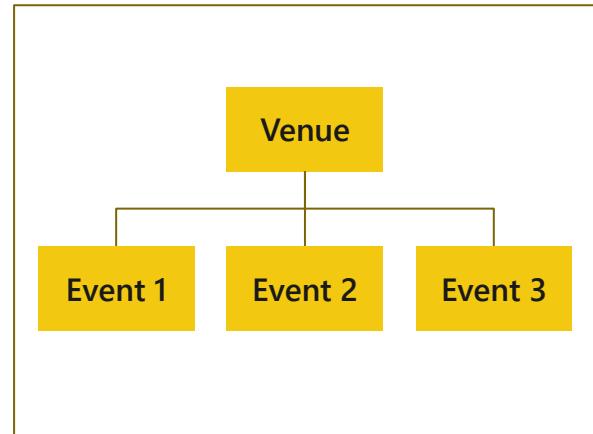
Relationship types – One-to-many (1:N)

One record of a primary table

Zero, one or many records of a related table

Related table has a lookup column

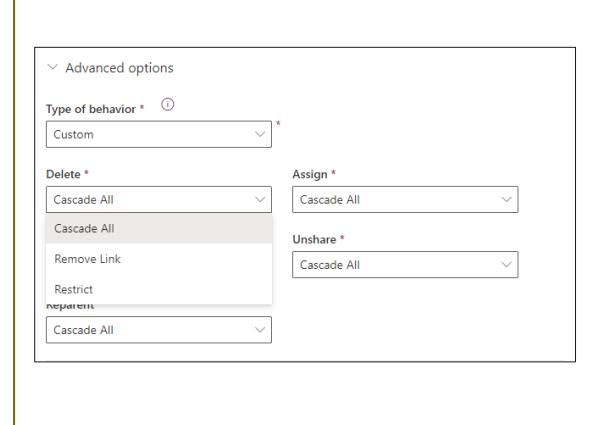
N:1 is a reverse 1:N relationships



© Copyright Microsoft Corporation. All rights reserved.

55

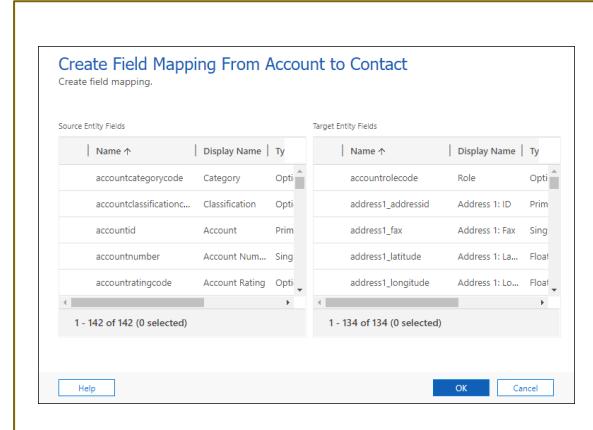
Relationship behaviors

 Preserve data integrity	 Rollup activity view	
 Parental relationships	 Automate business processes	

© Copyright Microsoft Corporation. All rights reserved.

56

Mappings

Operate on one-to-many relationships	
One-time only copy	
Used in model-driven app forms when creating a row from a sub-grid in the form	
Can use the InitializeFromRequest message in the API	

© Copyright Microsoft Corporation. All rights reserved.

57

Relationship types – Many-to-many (N:N)

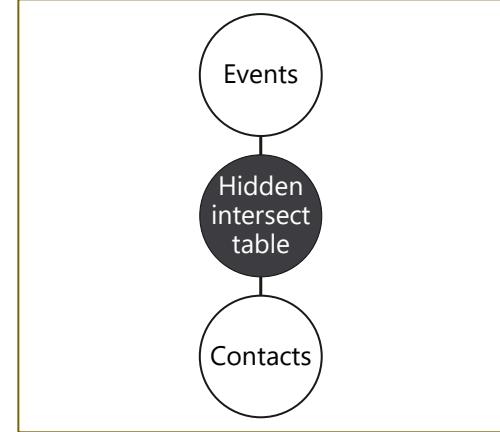
Many-to-many (N:N) is symmetrical

Hidden intersect table created

Two One-to-many relationships

Intersect table cannot be customized

- Can be used by report designers and developers



© Copyright Microsoft Corporation. All rights reserved.

58

Relationship types – N:N (connections)

Defines a connection between table rows without creating a formal relationship

Connection roles:

- Name
- Connection role category
- Description

The screenshot shows the 'New Connection' dialog in a software application. At the top, there are buttons for Save, Save & Close, New, and Flow. Below that, it says 'CONNECTION' and 'New Connection'. On the right, it shows 'Connected From' set to 'Maria Campbell (sample)' and 'As this role' set to 'Former Employee'. Under the 'Details' tab, there are fields for 'Connected From' (set to 'Maria Campbell (sample)'), 'Start Date' (set to 1/01/2015), and 'End Date' (set to 1/01/2018). There are also 'Save' and 'Close' buttons at the bottom.

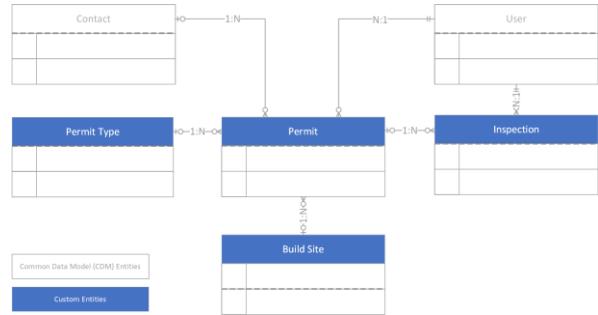
© Copyright Microsoft Corporation. All rights reserved.

59

Learning Path 1 practice labs

Data Modeling

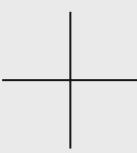
Lab 1. Data modelling



© Copyright Microsoft Corporation. All rights reserved.

60

Module 7: Create and define calculated and rollup columns in Microsoft Dataverse



© Copyright Microsoft Corporation. All rights reserved.

61

Microsoft Learn module

Create and define calculation or rollup columns in Dataverse

<https://learn.microsoft.com/training/modules/create-define-calculation-rollup-fields>



© Copyright Microsoft Corporation. All rights reserved.

62

Calculated and rollup columns

Simple column:

Contains data not based on formula

Example: First name, birthday, or email address

Calculated column:

Contains calculations that use columns from the current table or related parent table

Example: Full name (= First name + last name)

Rollup column:

Contains an aggregate value computed from the rows related to a row in a one-to-many relationship

Example: Potential revenue (= Sum of revenues of all open opportunities for a given account)

Behavior

Simple
Simple
Calculated
Rollup

© Copyright Microsoft Corporation. All rights reserved.

63

Calculated and rollup column data types

Calculated column

- Single line of text
- Choice
- Yes/No
- Whole number
- Decimal number
- Currency
- Date Time

Rollup column

- Whole number
- Decimal number
- Currency
- Date Time

© Copyright Microsoft Corporation. All rights reserved.

64

Calculated and rollup column limitations

Calculated columns

- Calculated on retrieve
- Can use columns on table and columns in many-to-one relationships

Rollup columns

- Max 100 per environment
- Max 10 per table
- Recalculated every hour
- Requires a one-to-many relationship

© Copyright Microsoft Corporation. All rights reserved.

65

Calculated and rollup column limitations



Values are read-only



Values not audited



Do not trigger workflow, code, or Power Automate cloud flows



Calculated columns cannot contain a calculated column from another table that also contains another column from a different table (spanning three tables)

© Copyright Microsoft Corporation. All rights reserved.

66

Module 8: Define and create business rules in Microsoft Dataverse



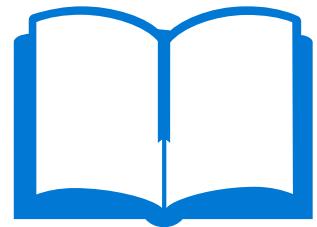
© Copyright Microsoft Corporation. All rights reserved.

67

Microsoft Learn module

Define and create business rules in Dataverse

<https://learn.microsoft.com/training/modules/define-create-business-rules>



© Copyright Microsoft Corporation. All rights reserved.

68

What behavior is expected when the logic happens on the server?



© Copyright Microsoft Corporation. All rights reserved.

69

Business rules

You can create business rules and recommendations to apply logic and validations without writing code



Automate and enforce common requirements



Rules are created in the context of a table



Rules can execute on both client and server



Provide an alternative to developer code-based rule implementation

© Copyright Microsoft Corporation. All rights reserved.

70

Define the components of a business rule

Business rules encapsulate logic in a predefined set of steps that will run each time that data is entered or modified, and the data meets the criteria to trigger the business rule



Condition – Business rules start with a condition



Action – Logic



Scope – defines where a business rule is applied

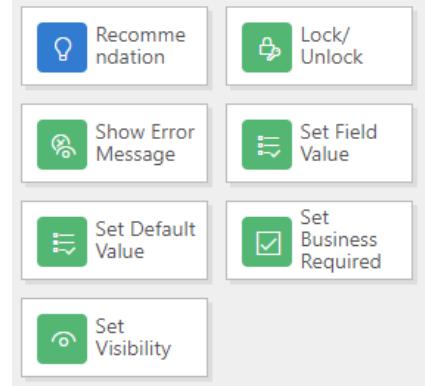
© Copyright Microsoft Corporation. All rights reserved.

71

Define the components of a business rule

Actions

- Set column values
- Clear column values
- Set column requirement levels
- Show or hide columns
- Enable or disable columns
- Validate data and show error messages
- Create business recommendations



© Copyright Microsoft Corporation. All rights reserved.

72

Define the components of a business rule

Scope

A business rule only applies in a canvas app when the scope set to **Entity**.

Scope of a business rule	Applies to
Entity (Table)	Model-driven app forms and server
All forms	All model-driven app forms
Specific form (Account form, for example)	Just that model-driven app form

© Copyright Microsoft Corporation. All rights reserved.

73

Module 9: Manage security in Microsoft Dataverse



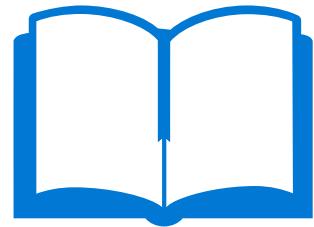
© Copyright Microsoft Corporation. All rights reserved.

74

Microsoft Learn module

Get started with security roles in Dataverse

<https://learn.microsoft.com/training/modules/get-started-security-roles>

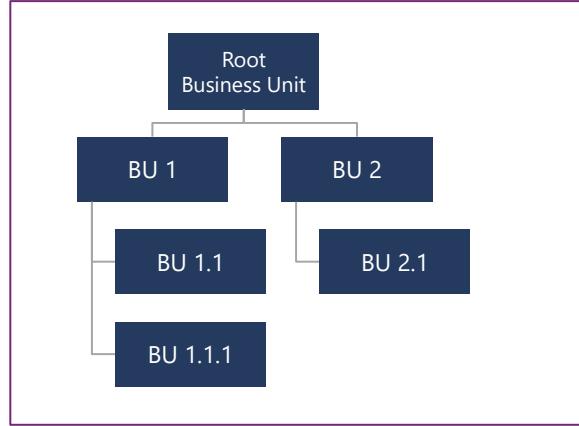


© Copyright Microsoft Corporation. All rights reserved.

75

Security model building blocks

- Teams
- Business units
- Security roles
- Column-level security
- Access teams
- Hierarchy security

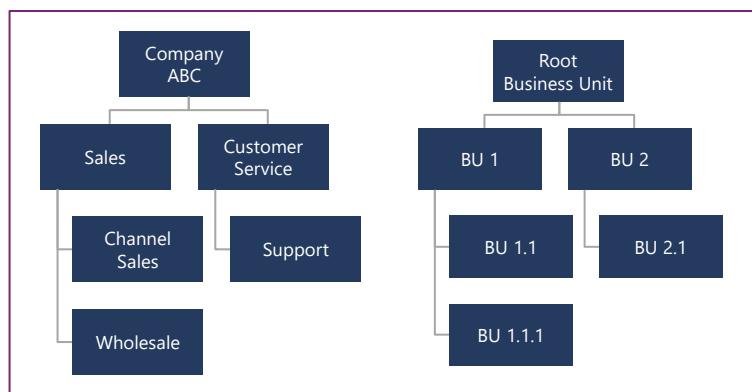


© Copyright Microsoft Corporation. All rights reserved.

76

Business units

Security boundaries for users (and teams) for access and privileges to organizational data



© Copyright Microsoft Corporation. All rights reserved.

77

Understand user security roles and security role defaults



Security roles are groups of permissions that you can assign to a user to grant them access and various capabilities and functionality like read, delete, or edit of records in a table within an environment



Security roles are either :

- Standard security roles created with every instance of Microsoft Dataverse
- Custom security roles created by an administrator

© Copyright Microsoft Corporation. All rights reserved.

78

Standard security roles



You can manage **Dataverse** security by using roles and then adding users to the environment and assigning roles to users



The built-in roles available to assign to users:

- System Administrator
- System Customizer
- Environment Maker
- Basic User
- App Opener

© Copyright Microsoft Corporation. All rights reserved.

79

Security roles overview

The screenshot shows a security role matrix for the 'Account Manager' role. The columns represent entities such as Account, Activity, Application File, etc., and the rows represent permissions like Create, Read, Write, Delete, Append, Append To, Assign, and Share. The matrix uses color coding to indicate the scope of each permission: User (yellow), Business unit (orange), Parent Child Business units (green), and Organization (dark green).

	Create	Read	Write	Delete	Append	Append To	Assign	Share
Account	User	User	User	User	User	User	User	User
Activity	User	User	User	User	User	User	User	User
Application File	User	User	User	User	User	User	User	User
Category	User	User	User	User	User	User	User	User
Connection	User	User	User	User	User	User	User	User
Connection Role	User	User	User	User	User	User	User	User
Contact	User	User	User	User	User	User	User	User
Customer Relationship	User	User	User	User	User	User	User	User
Data Import	User	User	User	User	User	User	User	User
Data Map	User	User	User	User	User	User	User	User
Data Performance Dashboard	User	User	User	User	User	User	User	User
Document Location	User	User	User	User	User	User	User	User
Document Suggestions	User	User	User	User	User	User	User	User
Duplicate Detection Rule	User	User	User	User	User	User	User	User
Email Signature	User	User	User	User	User	User	User	User
Email Template	User	User	User	User	User	User	User	User
Feedback	User	User	User	User	User	User	User	User
Follow	User	User	User	User	User	User	User	User
Import Source File	User	User	User	User	User	User	User	User
Interaction for Email	User	User	User	User	User	User	User	User

© Copyright Microsoft Corporation. All rights reserved.

80

Security roles in Power Platform admin center

The screenshot shows the security roles section in the Power Platform Admin Center. It lists various tables and their associated privileges for different user groups (User, Business unit, Parent Child Business units, Organization). The interface includes a search bar, a compact grid view toggle, and a 'Show only assigned tables' dropdown.

Table	Name	Record owner...	Permission...	Create	Read	Write	Delete	Append	Append to	Assign	Share
Business Management (17)	ACViewMapper	... acviewmapper	Organization	None	Organization	None	None				
Business Process Flows (3)	Account	... account	User or Team	Custom	User	User	User	User	User	User	Organization
Core Records (44)	Action Card	... actioncard	User or Team	Custom	User	User	User	User	User	Organization	None
	Action Card User Settings	... actioncardusersettings	User or Team	Private	User	User	User	User	User		
	Activity	... activitypointer	User or Team	Custom	User	User	User	User	User	User	Organization

© Copyright Microsoft Corporation. All rights reserved.

81

Privileges and access levels

Privileges:

- Create
- Read
- Write
- Delete
- Append
- Append to
- Assign
- Share

Access levels:

- User
- Business Unit
- Parent-Child Business Unit
- Organization

© Copyright Microsoft Corporation. All rights reserved.

82

Types of teams

Owner teams:

- Rows owned by groups of users instead of an individual user
- Number of teams is known at design time
- Reporting on progress by teams is required

Access teams:

- Teams need to be dynamically formed and dissolved
- Number of teams is not known at design time
- Team members require different access rights to rows
- Some users require access to a single row without requiring ownership

The screenshot shows the 'Managers' section of a Microsoft Teams page. At the top, there's a 'TEAM' dropdown and a 'Managers' button. Below that is a 'General' section with the following data:

Team Name *	HR Managers
Business Unit *	HR
Administrator *	Sven Mortensen
Team Type *	Owner

To the right of this is a 'Team members' sidebar with a search bar and a list of members:

- Search for records
- Full Name ↑
- Sven Mortensen
- William Contoso

© Copyright Microsoft Corporation. All rights reserved.

83

Azure Active Directory group teams to manage user app and data access



An Azure Active Directory (AAD) group team, similar to an owner team, can own records and can have security roles assigned to the team.



The administration of app and data access for Microsoft Dataverse has been extended to allow administrators to use their organization's Azure Active Directory (Azure AD) groups to manage access rights for licensed Dataverse users.

© Copyright Microsoft Corporation. All rights reserved.

84



© Copyright Microsoft Corporation. All rights reserved.

85

Learning Path 2: Create model-driven apps

© Copyright Microsoft Corporation. All rights reserved.



86

Module 1: Get started with model-driven apps



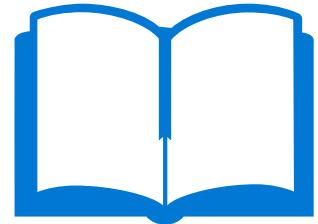
© Copyright Microsoft Corporation. All rights reserved.

87

Microsoft Learn module

Get started with model-driven apps in Power Apps

<https://learn.microsoft.com/training/modules/get-started-with-model-driven-apps-in-powerapps>



© Copyright Microsoft Corporation. All rights reserved.

88

What kind of apps have you made or used?



© Copyright Microsoft Corporation. All rights reserved.

89

Types of Power Apps

Model-driven

- Targets internal facing users
- Designed around the data
- Powered by Dataverse
- Automation
- Customizable user experience
- Customizable with or without professional developers

Canvas

- Targets internal facing users
- Designed around the user experience
- Connect to hundreds of data sources
- Customizable with or without professional developers

Portal (Power Pages)

- Targets external stakeholders
- Public-facing website
- Powered by Dataverse
- Customizable with or without professional developers

© Copyright Microsoft Corporation. All rights reserved.

90

Model-driven apps

- List and forms over data
- Consistent experience across apps and devices
- Easy to build with configuration first approach
- Model-driven apps require Microsoft Dataverse
- You can build more than one model-driven app from a single Dataverse environment

The screenshot shows a Microsoft Power Apps portal page titled "Connected Operations Challenge". At the top, there's a summary card for "Connected Operations" with a value of 5, a due date of 4/29/2018, and a status of "Active". Below this is a "Challenge Management" section with a red button labeled "Active for 5 days". The main area displays a table titled "Active Challenges" with two rows: "Connected Operations" and "Enterprise sustainability". A large callout box highlights the "Setup" and "Track (5 D)" sections of the challenge card, which include fields for "Domain" (Industry IOT), "Duration (weeks)" (10), "Additional Stakeholders" (checkboxes for "Complete" and "Launch Challenge"), and a "Set Active" button.

© Copyright Microsoft Corporation. All rights reserved.

91

Built-in to all model-driven apps

Advanced Find	Searching	Personal Views	Personal Dashboards
Editable Grids	Excel Integration	Client Dev Extensibility	Client-side Business rules
Reusable Quick View Forms	Business Process Control	Timeline Activity View	Command Bar Extensibility
Duplicate Detection	Multiple Makers	Recently Viewed/Pinned	Localization support

© Copyright Microsoft Corporation. All rights reserved.

92

App building blocks



Data: Define the data the app will be based upon



User interface: Determine how users interact with the app



Logic: Determine what business processes, rules, and automation the app will have



Visualization: Determine what type of data and reporting the app will show

© Copyright Microsoft Corporation. All rights reserved.

93

User interface

Determine how users will interact with the app:



App: Apps determine the app fundamentals, like components, properties, and the URL



Site Map: Specifies the navigation of your app



Form: Include a set of data entry columns for a given table



View: Defines how a list of rows for a specific table appear in your app

© Copyright Microsoft Corporation. All rights reserved.

94

Visualization

Determine what type of data and reporting the app will show:



Chart: Individual graphical visualizations that can appear in a view or a form or can be added to a dashboard



Dashboard: Show one or more graphical visualizations that provide an overview of actionable business data



Embedded Microsoft Power BI: Adds embedded Power BI tiles and dashboards to your app

© Copyright Microsoft Corporation. All rights reserved.

95

Control security when sharing model-driven apps

Model-driven apps use role-based security for sharing



Access to the app



Access to tables in the app



Access to model-driven app features

© Copyright Microsoft Corporation. All rights reserved.

96

Module 2: Configure forms



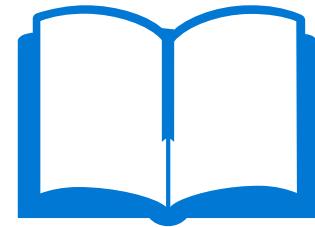
© Copyright Microsoft Corporation. All rights reserved.

97

Microsoft Learn module

Configure forms, charts, and dashboards in model-driven apps

<https://learn.microsoft.com/training/modules/configure-model-driven-apps-customer-engagement-apps>



© Copyright Microsoft Corporation. All rights reserved.

98

Form designer

Model-driven forms are how data for a row is presented to the user during editing and viewing.

The screenshot shows the Microsoft Dynamics 365 Form Designer. On the left, there's a tree view of form sections: Tree view, Search, Information, General, Details (selected), Building Name, Country, Date Built, Value, Space, Floor, and Footer. The main workspace shows a 'New Building' form with fields for Building Name, Country, Date Built, and Space. The right pane is the Properties panel for the 'Building Name' field, showing settings for Single-line text, Display options, Table column, Label, and Formatting.

© Copyright Microsoft Corporation. All rights reserved.

99

Form structure

Model-driven app forms have a defined structure



Tabs



Sections



Controls

© Copyright Microsoft Corporation. All rights reserved.

100

Form component controls

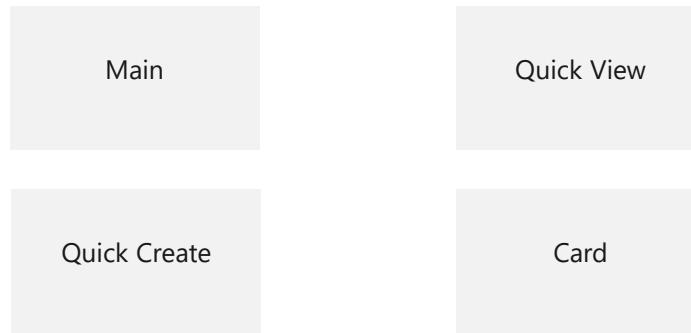
Components can be added to columns on the forms to display how the control is displayed

The screenshot shows the Power Apps Form Builder interface. On the left, the 'Components' pane is open, displaying a tree view of available controls under categories like Layout, Display, Input, and More. A red box highlights the 'Input' category. On the right, a configuration dialog for 'Add Rich Text Editor Control' is shown. It includes fields for 'Table column *' (set to 'healthcare_feedback (Text)'), 'Field (1)' (set to 'Rich Text Editor Control'), and a list of other component options like 'HTML web resource', 'Image web resource', and 'Text (Multiline Text)'. A red box highlights the 'Rich Text Editor Control' option in the list. At the bottom of the dialog, there is a 'Show component on' button.

© Copyright Microsoft Corporation. All rights reserved.

101

Form types



© Copyright Microsoft Corporation. All rights reserved.

102

Specialized form components

- Timeline
- Sub-grid
- Quick view
- Reference panel
- Timer control
- IFrame

The screenshot displays the Microsoft Dynamics 365 form builder interface. On the left, there is a large panel titled 'Timeline Properties' containing several sections: 'Display options' (Name: Timeline, Record types to show: Activities, Notes), 'Profile picture' (Default), 'Advanced' (Quick entry record type: Notes, Sort default: Descending, Display layout: Roomy, Records shown on page: 10), and 'Additional settings' (Filter pane, Hide Timeline label, Hide the date portion for any datetime in the past 24 hours, Enable search bar, Expand all records by default, Enable "What you've missed" summary, Move timeline to a new tab for mobile apps, Default). To the right of this panel are three smaller sections: 'Activities' (Appointment, Email, Fax, Letter, Phone Call, Recurring Appointment, Social Activity, Task, Teams chat), 'Sort activities by' (Last Updated, Create activities using: Quick create form, Open activities using: Main form, Activity rollup type: Extended), and 'Additional settings' (Enable filter pane, Expand filter pane by default, Edit filter pane).

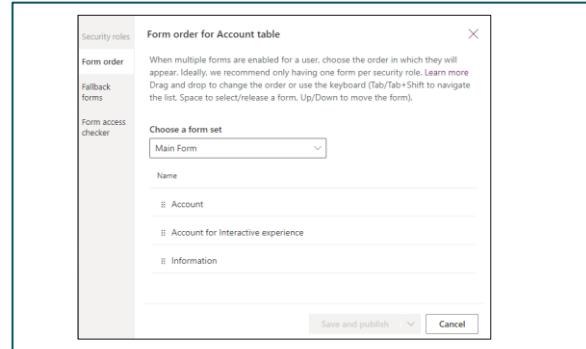
© Copyright Microsoft Corporation. All rights reserved.

103

Configure multiple forms

Users of model-driven apps will encounter many forms in their user experience, and will even encounter different forms for the same table depending on their role

You can create more than one Quick Create form for a given table, but only the first one in order will be available for end users.



© Copyright Microsoft Corporation. All rights reserved.

104

Access to forms



Each table must have a designated fallback form. This is the form that all users would see if they do not have a security role assigned that matches your role tailored forms.



Only Main forms can be assigned to security roles.



When a user has access to multiple forms, a form selector will be available near the top of the form



You can designate a main form as inactive. This will make it not visible to all users, regardless of security roles.

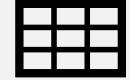


The table definition must allow for Quick Create before you will be able to configure a Quick Create form

© Copyright Microsoft Corporation. All rights reserved.

105

Module 3: Configure views



© Copyright Microsoft Corporation. All rights reserved.

106

Overview of views

Views are lists of data from a Dataverse table

- **Columns from table**
- **Columns from many-to-one relationships**
- **Sorting**
- **Filtering**

Building Name	Country	Number of...	Available F...	Owner
Building A	Sweden	3	300	[Redacted]
Building B	Denmark	1	25	[Redacted]
Building C	Sweden	1	200	[Redacted]

© Copyright Microsoft Corporation. All rights reserved.

108

Public vs Personal views

Public (System) views are managed in the Maker portal

- Public views are not controlled by security
- Public views can be turned off
- Public views can be included/excluded in an app

Personal (My) views are created in a model-driven app

- Personal views can be shared with users and teams

The screenshot shows the Dynamics 365 Maker portal interface. At the top, there's a navigation bar with 'Tables > Building > Views'. Below it is a list of views for the 'Building' entity, each with a name, description, and a 'View type' dropdown. One view, 'Building Details', is highlighted. To the right of the list is a sidebar titled 'Building Details' with sections for 'My Views' and 'System Views', and a button to 'Set current view as my default'.

© Copyright Microsoft Corporation. All rights reserved.

109

Specialty views

There are four types of specialty view; **Lookup, Associated, Quick Find, Advanced Find**

The screenshot shows the Dynamics 365 model-driven app interface. On the left, there's a 'Contacts' list view with columns for 'Full Name' and 'Business Phone'. On the right, there's a 'Related - Common' section listing various entities such as Activities, Social Profiles, Contacts, Connections, Entitlements, Actuals, Resource Preferences, Opportunity Lines, Quotes, and Quote Lines.

© Copyright Microsoft Corporation. All rights reserved.

110

Module 4: Command bar



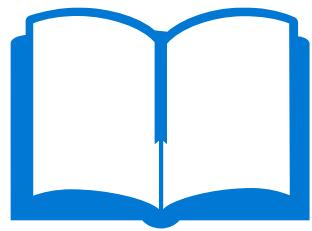
© Copyright Microsoft Corporation. All rights reserved.

111

Microsoft Learn module

Customize the command bar

<https://learn.microsoft.com/training/modules/command-bar-customize>



© Copyright Microsoft Corporation. All rights reserved.

112

Command bar

Locations



Main grid



Main form



Subgrid view



Associated view

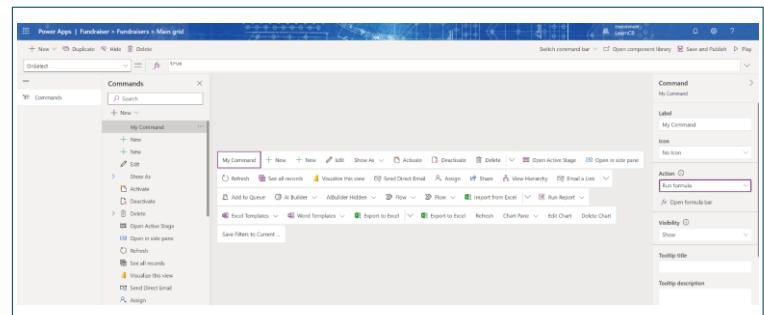
© Copyright Microsoft Corporation. All rights reserved.

113

Command bar designer

Edit in model-driven app

- Choose command bar
- Visual representation
- Add new commands
- Modify existing commands



© Copyright Microsoft Corporation. All rights reserved.

114

Configuring actions

Power Fx

- **Properties**
 - OnSelect
 - Visible
- **Example use cases**
 - Access related rows
 - Create and update rows with Patch

JavaScript

- Requires JavaScript web resources
- Covered later in the course

© Copyright Microsoft Corporation. All rights reserved.

115

Learning Path 2 practice labs

Power Apps

Lab 2: Model driven app



© Copyright Microsoft Corporation. All rights reserved.

116



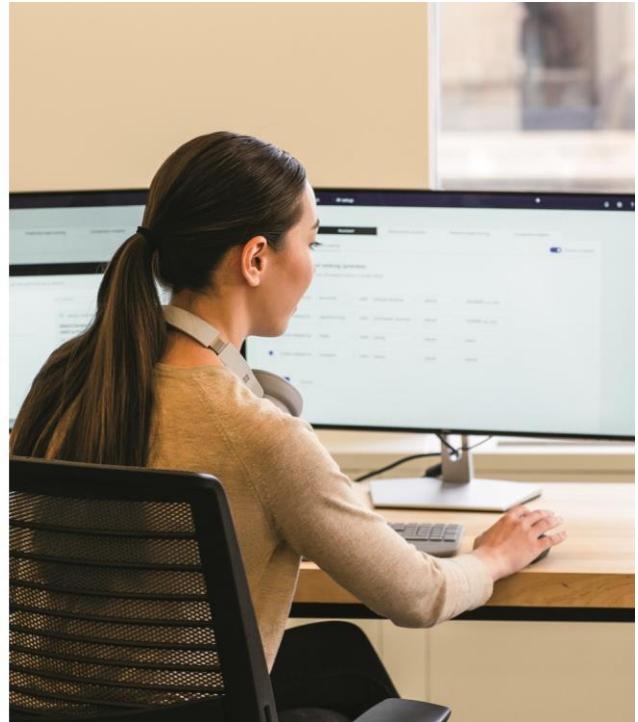
© Copyright Microsoft Corporation. All rights reserved.

117



Create canvas apps

© Copyright Microsoft Corporation. All rights reserved.



118

Module 1: Get started with Power Apps



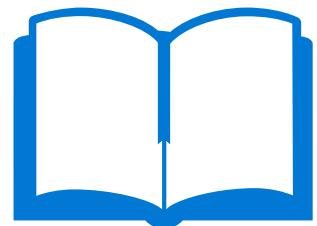
© Copyright Microsoft Corporation. All rights reserved.

119

Microsoft Learn module

Get started with Power Apps canvas apps

<https://learn.microsoft.com/training/modules/get-started-with-model-driven-apps-in-powerapps>



© Copyright Microsoft Corporation. All rights reserved.

120

Introduction to canvas apps

Power Apps is a suite of apps, services, connectors, and a data platform that provides you with an opportunity to build custom apps for your business needs.

The diagram consists of three main sections:

- Connect to data & systems you're already using easily:** Represented by a cloud icon with a gear and a database icon.
- Create apps, forms, and workflows *without writing code*:** Represented by a computer monitor icon showing a form.
- Use apps *on any device* – both web and mobile:** Represented by a smartphone and a tablet icon.

To the right of the diagram is a list of benefits:

- Build an app quickly by using the skills that you already have
- Connect to the cloud services and data sources that you're already using
- Share your apps instantly so that coworkers can use them on their phones and tablets

© Copyright Microsoft Corporation. All rights reserved.

121

Canvas apps building blocks

Tools and applications used



Power Apps maker portal - Apps start here, whether you build them from data, a sample app, or a blank screen.



Power Apps Studio - Create and edit apps



Power Apps mobile - Run your apps on Microsoft Windows, Apple iOS, and Google Android devices.

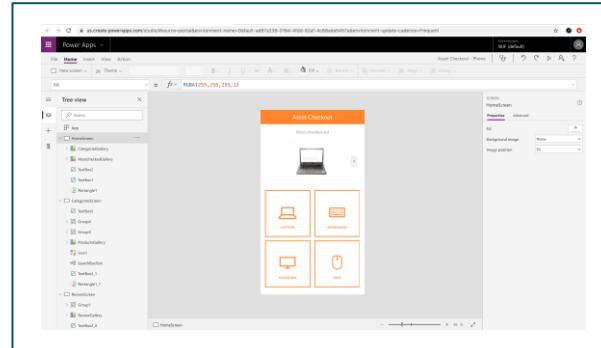
© Copyright Microsoft Corporation. All rights reserved.

122

Power Apps Studio

Power Apps Studio is where you can fully develop your apps to make them more effective as a business tool and to make them more attractive.

- **Left pane** - Shows a hierarchical view of all the controls on each screen or a thumbnail for each screen in your app
- **Middle pane** - Shows the canvas app that you're working on
- **Right pane** - Where you set options such as the layout, properties, and data sources for certain controls



© Copyright Microsoft Corporation. All rights reserved.

123

Module 2: Understanding Low Code as a Traditional Developer



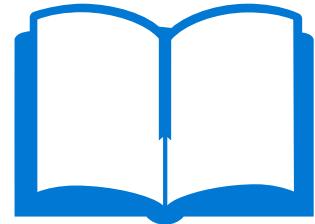
© Copyright Microsoft Corporation. All rights reserved.

124

Microsoft Learn module

Understanding Low Code as a Traditional Developer

<https://learn.microsoft.com/training/modules/understanding-low-code-as-a-traditional-developer>



© Copyright Microsoft Corporation. All rights reserved.

125

What is low code?

Write small amount of code that would normally takes several more lines of code in a traditional programming environment

Power Fx formulas which have an Excel like syntax

Power Fx is a declarative language

Power Fx formula:

```
fx "Employee: " & LookUp( Orders, Number=Selected ).Employee.'First Name'  
Employee: Elisa
```

JavaScript equivalent requires explicit: **Asynchronous and lambda**

```
1 RunQueryAsync(Orders,  
2   "$filter=nwind_ordernumber+eq+" + Escape(Selected) +  
3   "&$expand=nwind_EmployeeID($select=nwind_employeeid,nwind_firstname)" +  
4   "&$select=nwind_employeesid,nwind_ordernumber,nwind_EmployeeID,_nwind_employeeid_value" +  
5   "&$top=1"  
6   ).then(  
7     (result) => "Employee: " + result.nwind_firstname  
8   );
```

© Copyright Microsoft Corporation. All rights reserved.

126

Module 3: Customize a canvas app



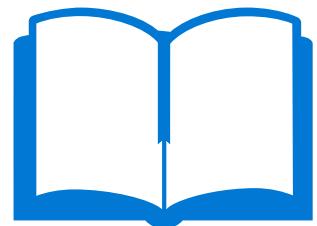
© Copyright Microsoft Corporation. All rights reserved.

127

Microsoft Learn module

Customize a canvas app in Power Apps

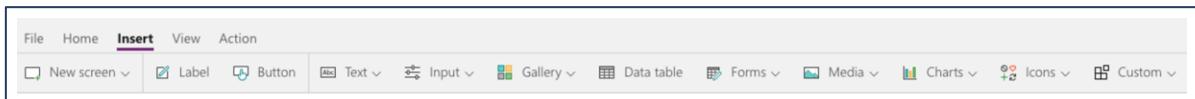
<https://learn.microsoft.com/training/modules/customize-apps-in-powerapps>



© Copyright Microsoft Corporation. All rights reserved.

128

Controls in Power Apps



UI elements that produces an action or shows information

Common controls include:

Galleries:

Layout containers that hold a set of controls that show rows from a data source

Forms:

Show details about your data and let you create and edit rows

Label:

A label shows data that you specify as a literal string of text, which appears exactly the way you type it, or as a formula that evaluates to a string of text

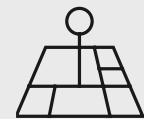
Text Input:

A box in which the user can type text, numbers, and other data

© Copyright Microsoft Corporation. All rights reserved.

129

Module 4: Navigation in canvas apps



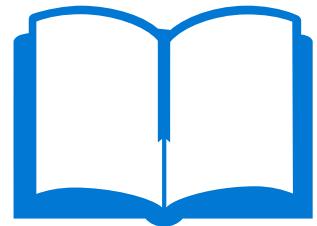
© Copyright Microsoft Corporation. All rights reserved.

130

Microsoft Learn module

Navigation in a canvas app in Power Apps

<https://learn.microsoft.com/training/modules/navigation-canvas-app>



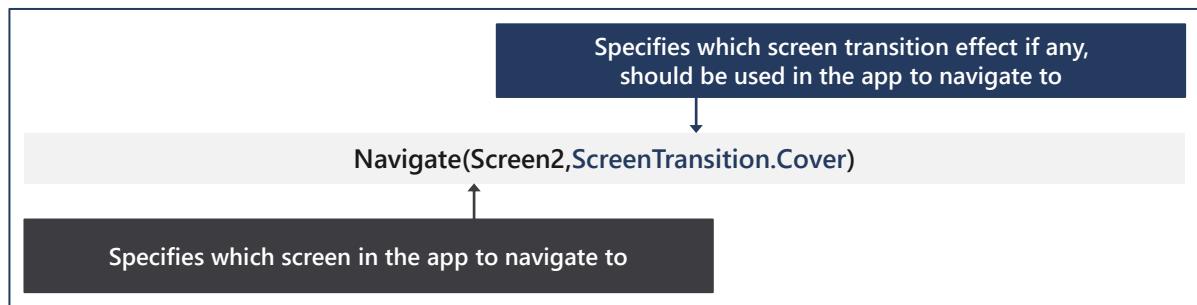
© Copyright Microsoft Corporation. All rights reserved.

131

Understanding navigation

Since most apps have multiple screens, understanding how to implement the Navigate function in your app is critical

Typically added to the OnSelect property of a Button or Icon control



© Copyright Microsoft Corporation. All rights reserved.

132

Module 5: Power Fx formulas



© Copyright Microsoft Corporation. All rights reserved.

133

Microsoft Learn module

Create formulas to change properties in a Power Apps canvas app

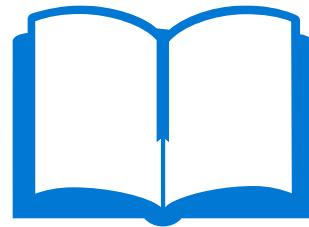
<https://learn.microsoft.com/training/modules/author-basic-formula-change-properties-powerapps>

Create formulas to change behaviors in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/author-basic-formula-change-behaviors-powerapps>

Author a basic formula that uses tables and records in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/author-basic-formula-tables-records-powerapps>



© Copyright Microsoft Corporation. All rights reserved.

134

Power Fx formulas

When using Microsoft Power Apps, you don't have to write complicated application code the way that a traditional developer does.

Configure your app with formulas that not only calculate values and perform other tasks (as they do in Excel) but also respond to user input

For example, you build a formula to determine how your app responds when users select a button, adjust a slider, or provide other input.



© Copyright Microsoft Corporation. All rights reserved.

135

Power Fx

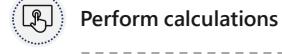
What can formulas be used for in canvas apps



Set values



Format output



Perform calculations



Control visibility and display mode of controls



Set the position of controls



Filter data



And much much more

© Copyright Microsoft Corporation. All rights reserved.

136

Core properties of controls

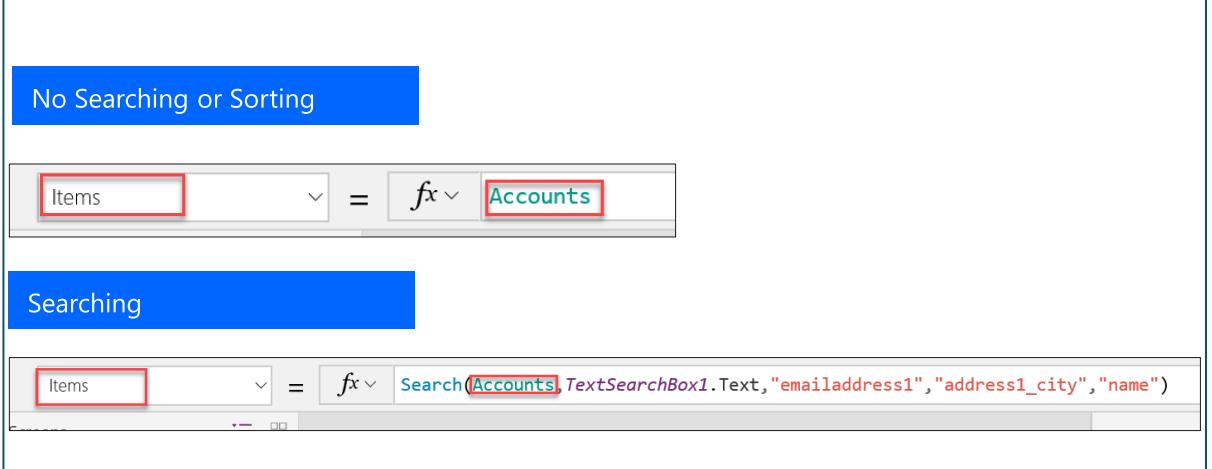
While each control may have slightly different properties the following are important ones to be aware of:

-  **Default:** Initial value of control before it is changed by the user
-  **Items:** The source of data that appears in a control such as a gallery, list, or chart
-  **OnChange:** How the app responds when the user changes the value of a control. For example, when a user selects a different value in a Dropdown control
-  **OnSelect:** How the app responds when the user taps or clicks a control
-  **Text:** Text that appears on a control or that the user types into a control
-  **Tooltip:** Explanatory text that appears when the user hovers over a control
-  **Visible:** Whether a control appears or is hidden

© Copyright Microsoft Corporation. All rights reserved.

137

Connecting the Gallery to Data



No Searching or Sorting

Items = Accounts

Searching

Items = Search(Accounts, TextSearchBox1.Text, "emailaddress1", "address1_city", "name")

© Copyright Microsoft Corporation. All rights reserved.

138

Filtering Data

- Filter function applies the formula to each record
- Can include =,<>,<,>=,<=,&&,&|| and in operators

The screenshot shows the Power Query Editor interface with four examples of filter formulas:

- Top-left: `Filter(Accounts, Status = 'Status (Accounts)'.Active)`
- Top-right: `Filter(Accounts, Status = 'Status (Accounts)'.Active, 'Account Name' = 'Contoso')`
- Middle-left: `Filter(Accounts, Status = 'Status (Accounts)'.Active, 'Account Name' = "Contoso" & 'Account Name' = "Fabrikam")`
- Middle-right: `Filter(Accounts, Status = 'Status (Accounts)'.Active, 'Account Name' = "Contoso" || 'Account Name' = "Fabrikam")`

Below the formulas, the M code editor shows the expanded logic:

```

Search(
    Filter(
        Accounts,
        Status = Active,
        'Account Name' = "Contoso" || 'Account Name' = "Fabrikam"
    ),
    searchText.Text,
    "name"
)
  
```

Buttons at the bottom: Format text, Remove formatting.

© Copyright Microsoft Corporation. All rights reserved.

139

Using App checker

Indicates in real-time if an app contains errors

Clicking on the item will take you to that formula

Clicking on the > on the item will take you to the details of the error

The screenshot shows the App checker interface with two main sections:

- No Errors**: Shows a message indicating there are no errors.
- Errors Detected**: Shows a message indicating errors are detected.

A central workspace displays a Power Automate flow named "rice Ordering App". A red box highlights the "U" icon in the top right corner of the workspace, which links to the error details.

To the right, two floating panes provide detailed information about the errors:

- Formulas** pane: Shows a list of errors under "MainScreen (2)". One error is highlighted: "DeviceGallery" with the message "Name isn't valid. This identifier isn't recognized." A red box highlights the "DeviceGallery" link.
- Details** pane: Shows the specific issue: "Name isn't valid. This identifier isn't recognized. This error appears most commonly when a formula refers to something that no longer exists (for example, a control that you've deleted)." It also includes a "Location" section pointing to "DeviceGallery.Items" and a "How to fix" section.

This tool also finds potential accessibility issues and explains why each might be a potential problem

© Copyright Microsoft Corporation. All rights reserved.

140

Module 6: Canvas components



© Copyright Microsoft Corporation. All rights reserved.

141

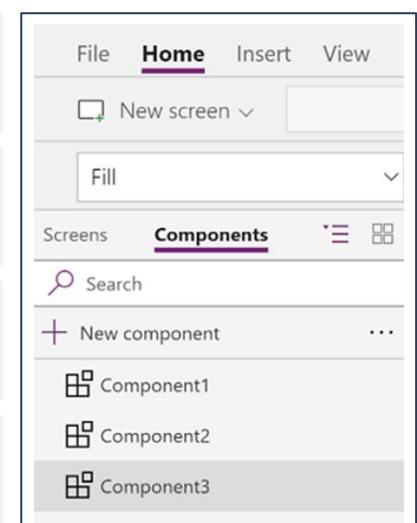
Components

Reusable building blocks

Can be used multiple times

Custom properties define interface with hosting screen

Component libraries



© Copyright Microsoft Corporation. All rights reserved.

142

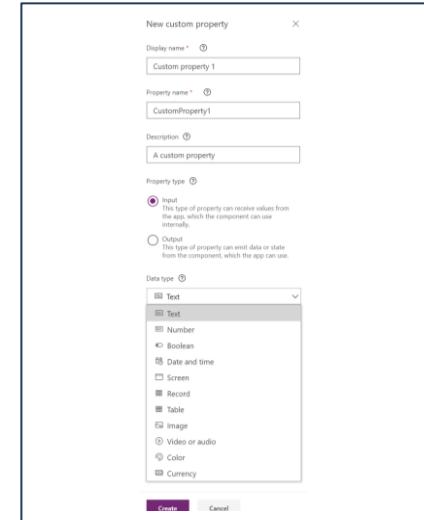
Component custom properties

Can be Input or Output

Default value is assigned based on data type

Enable Access app scope to share with component:

- Global variables
- Collections
- Controls on screens
- Tabular data sources



© Copyright Microsoft Corporation. All rights reserved.

143

Component Libraries

Repository of components for reusability

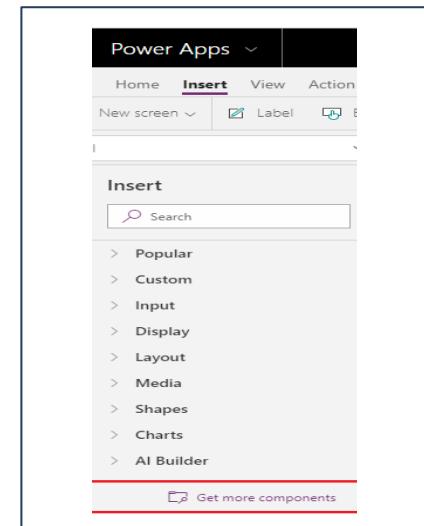
Discover and search components

Publish updates

Notify app makers of available component updates

Import from a component library

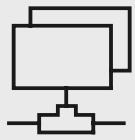
Notification in Power Apps Studio when an updated component in the library is available



© Copyright Microsoft Corporation. All rights reserved.

144

Module 7: Testing canvas apps



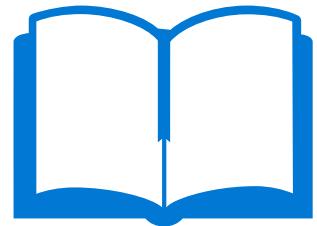
© Copyright Microsoft Corporation. All rights reserved.

145

Microsoft Learn module

Document and test your Power Apps application

<https://learn.microsoft.com/training/modules/document-test-powerapps-app>

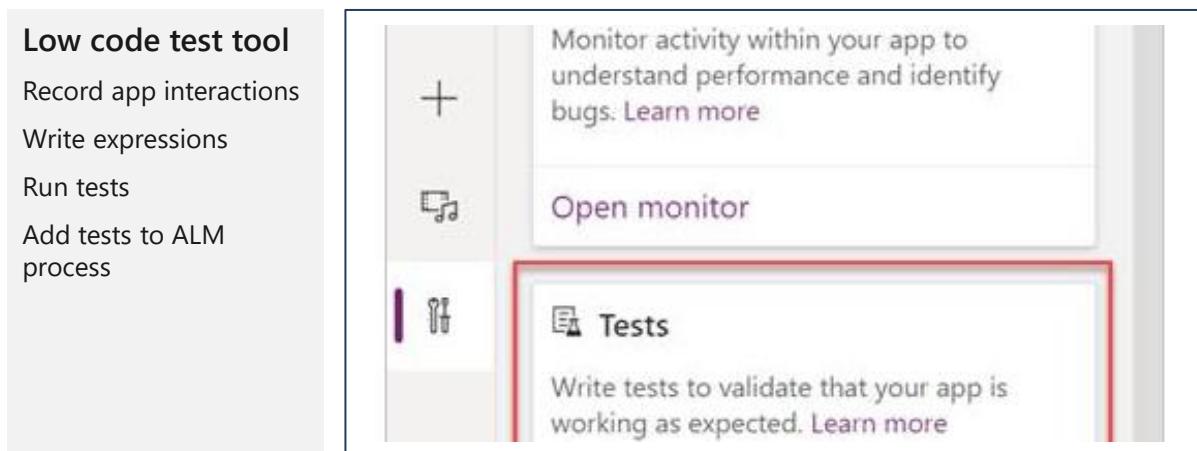


© Copyright Microsoft Corporation. All rights reserved.

146

Power Apps Test Studio

Automate the testing of canvas apps



147

Power Apps Test Studio

Components of Test Studio



© Copyright Microsoft Corporation. All rights reserved.

148

Learning Path 3 practice labs

Power Apps

Lab 3: Canvas app



© Copyright Microsoft Corporation. All rights reserved.

149

 Microsoft Power Platform

© Copyright Microsoft Corporation. All rights reserved.

150



Advanced techniques in canvas apps

© Copyright Microsoft Corporation. All rights reserved.



151

Module 1: Use imperative development techniques for canvas apps



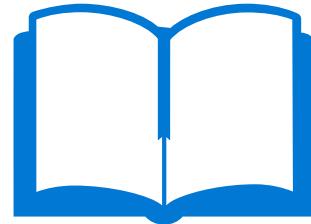
© Copyright Microsoft Corporation. All rights reserved.

152

Microsoft Learn module

Use imperative development techniques for canvas apps in Power Apps

<https://learn.microsoft.com/training/modules/use-imperative-dev-techniques-powerapps-canvas-app>



© Copyright Microsoft Corporation. All rights reserved.

153

The three types of variables in Power Apps

In your app, you can use variables.

- **Global variables** -- The most traditional type of variable. You use the **Set** function to create and set its value. Then you can reference its values anywhere within your app. A common use is to store a user's DisplayName when the app loads and then reference the variable throughout the app.
- **Context variables** -- A context variable is only available on the screen where you create it using the **UpdateContext** function. Context variables are commonly used for functionality that controls a pop-up screen, for example, where you want to use the same variable name on multiple screens but maintain the value separately.
- **Collections** -- A collection is a special type of variable for storing a table of data. You can create the collection manually or by loading another data sources table into it. Collections are available throughout your app, like global variables, and they are created using the **Collect** or **ClearCollect** function.

© Copyright Microsoft Corporation. All rights reserved.

154

Global variables

A common design pattern in apps is to personalize the app

```
"Welcome " & User().FullName
```

```
Set(varUserDisplayName, User().FullName)
```

```
"Welcome " & varUserDisplayName
```

© Copyright Microsoft Corporation. All rights reserved.

155

Tracking status in a variable

In a declarative mindset, you might hide or show controls based on a query for data.

```
CountRows(Filter(InvoiceTable, CustomerNumber = ThisCustomersNumber  
And Status = "Outstanding")) > 3
```

```
Set(varOutstandingExceeded, CountRows(Filter(InvoiceTable, CustomerNumber =  
ThisCustomersNumber And Status = "Outstanding")) > 3)
```

© Copyright Microsoft Corporation. All rights reserved.

156

Contextual variables

Contextual variables are similar to global variables except they are only referenced on the screen where you create them.

```
Set(varCount, 1);Set(varActive, true);Set(varName, User().FullName)
```

```
UpdateContext({varCount: 1, varActive: true, varName: User().FullName})
```

```
UpdateContext({varShowPopUp: true})
```

© Copyright Microsoft Corporation. All rights reserved.

157

Using collections to increase performance

The most common reason for using collections is to optimize performance by reducing calls to the same table in a data source.

- The Collect function is not delegable. This means by default only the first 500 rows from the data source will be retrieved and stored in the collection. For more information about working with delegation, see [Work with data source limits \(delegation limits\) in a Power Apps canvas app](#)
- Collections are not linked to the data source after you create them. This means changes to the data in the collection do not automatically save to the data source. If you want to update the data source based on your changes to the collection, you will need to build formulas to do so.
- Collections are temporary. When you close the app, the collection and all of its contents are removed. If you need to store collection data, you need to write it to a data source before closing the app.

`Collect(collectProjects, Projects)`

© Copyright Microsoft Corporation. All rights reserved.

158

A variable can store a single row

This concept applies to global and context variables.

```
Set(varUser, User())
```

```
varUser.Email
```

© Copyright Microsoft Corporation. All rights reserved.

159

Module 2: Perform custom updates in a canvas app



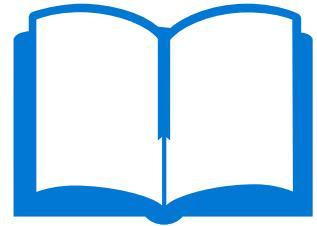
© Copyright Microsoft Corporation. All rights reserved.

160

Microsoft Learn module

Perform custom updates in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/perform-custom-updates-powerapps-canvas-app>



© Copyright Microsoft Corporation. All rights reserved.

161

Directly create and edit rows

In this module, you will learn about using the Patch function to create and update your data sources without the use of forms directly.

```
Patch(LoggingTable,  
    Defaults(LoggingTable),  
    { WhoClicked: User().FullName,  
      WhenClicked: Now() } );
```

© Copyright Microsoft Corporation. All rights reserved.

162

Using Patch to create a row

The Patch function can be used to create a new row in your data source.

- Include the name of the data source you want to edit. This could be a tabular data source (such as Dataverse or SharePoint) or a collection. For the example, you will use **CustomerOrders** as the name of the data source.
- The **Defaults** function returns a row that contains the default values for the data source. If a column within the data source doesn't have a default value, that property won't be present. By using Defaults with the data source, this notifies Patch to create a new row.
- Include the columns that you want to populate in the new row. Here you will specify the name of the column to update followed by the value to write to that column. For this example, you will update the Region and Country column with a string value.

```
Patch(CustomerOrders, Defaults(CustomerOrders), {Region: "Americas", Country: "Canada"})
```

© Copyright Microsoft Corporation. All rights reserved.

163

Using Patch to edit a row

It is also possible to edit a row in the data source.

- Include the name of the data source you want to edit. This could be a tabular data source (such as Dataverse or SharePoint) or a collection. For the example, you will use **CustomerOrders** as the name of the data source.
- The row that you want to edit in the data source. The most common way to specify this row is to use the **LookUp** function to retrieve the row from the data source. Another option if you are using a Gallery and you want to update the current row is to use the **ThisItem** function for referencing the row. For this example, you will use a LookUp function.
- Include the changes that you want to make. Here you will specify the name of the column to update followed by the value to write to that column. For this example, you will update the Region and Country column with a string value.

```
Patch(CustomerOrders, LookUp(CustomerOrders, ID = 1), {Region: "Asia", Country: "China"})
```

© Copyright Microsoft Corporation. All rights reserved.

164

Updating columns with Patch

The primary logic of most Patch functions is updating the proper columns with the correct information.

- Make sure you are updating all of the required columns from your data source.
- You can update as many or as few of the optional columns as you would like.
- Make sure your column names are spelled and capitalized correctly. Column names are case-sensitive.
- Make sure you are writing the correct data type. For example, if your column in the data source is a number type, then you cannot write a string value to it, even if that string contains a number.

165

Delete a row

There are also functions available for deleting one or more rows from your data source.

- **Remove and RemoveIf** - These functions are used to remove or delete rows from the data source.
- **Clear** - Use the **Clear** function to remove all of the rows from a collection.

```
Remove(CustomerOrders, LookUp(CustomerOrders, ID = 1))
```

© Copyright Microsoft Corporation. All rights reserved.

166

Delete all of the rows in a collection

The **Clear** function deletes all the rows of a collection.

- All at once - For example, if you are reloading the items in a collection for a drop-down menu when a screen becomes visible, you would want to use **ClearCollect**. One formula would then remove the old rows and add the new rows.
- Multi-step - For example, if you are using collections to store user inputs like in a shopping cart you can use **Clear** and **Collect**. This is because the user might want to clear their shopping cart without adding a new row.

```
Clear(collectSelectedItems)
```

© Copyright Microsoft Corporation. All rights reserved.

167

Bulk changes to rows

Patch and **Remove** are both functions that are used to affect one row

- Use the **ForAll** function, to loop through a table of data and run a Patch or Remove function for each row in the table.
- Use the **Collect** function to write from one table to another. Each row of the source table is added as a separate row to the target table.

168

Module 3: Use Dataverse choice columns with formulas



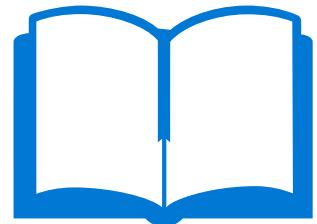
© Copyright Microsoft Corporation. All rights reserved.

169

Microsoft Learn module

Use Dataverse choice columns with formulas

<https://learn.microsoft.com/training/modules/choice-columns-formulas>



© Copyright Microsoft Corporation. All rights reserved.

170

Filter with Choice columns

Items in Choice columns are a key/value pair

- Value is stored on row
- Label is stored in metadata
- Labels are available in canvas app
- For Dataverse column can filter as follows

```
Filter(
  Accounts,
  Category = 'Category (Accounts)'. 'Preferred Customer'
)
```

© Copyright Microsoft Corporation. All rights reserved.

171

Filter using a Choice dropdown

Use Choices() function to display list of items from a choice column

Items = Choices(Accounts.Category)

Filter(Accounts, Category in ComboBoxCategory.SelectedItems)

Standard Preferred Customer

Fabrikam

City Power & Light (sample)

172

Patch with Choice columns

Choice columns require the choice set name when using Patch

```
Patch(Accounts, ThisItem, {Category: Category.'Preferred Customer'})
```

© Copyright Microsoft Corporation. All rights reserved.

173

Module 4: Work with relational data in a canvas app



© Copyright Microsoft Corporation. All rights reserved.

174

Microsoft Learn module

Work with relational data in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/work-with-relational-data-powerapps-canvas-app>

Reduce complexity in your data model with Dataverse table relationships

<https://learn.microsoft.com/training/modules/reduce-complexity-dataverse-table>



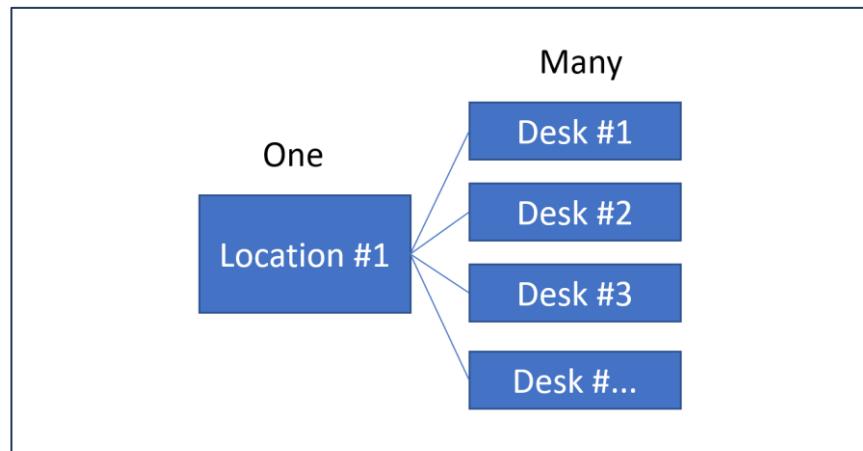
© Copyright Microsoft Corporation. All rights reserved.

175

One-to-many relationships

One-to-many relationships are the most common Dataverse relationships that you will work with.

The following example is a one-to-many relationship from the Location table to the Desk table



© Copyright Microsoft Corporation. All rights reserved.

176

Filter using a One-to-many relationship

Use Filter() function to display list of desks for a selected location

```
Items = Filter(Desks, Location.Location = FilterLocation_1.Selected.Location)
```

Note joining on the record not the Guid using the dot notation

177

Get related rows using a One-to-many relationship

Use the dot notation to retrieve related rows

```
Items = FilterLocation_1.Selected.Desks
```

178

Get row from a Many-to-one relationship

Use the dot notation on the lookup column to retrieve related row

```
Text = fx ▾ ThisItem.Location.Address
```

179

Module 5: Work with data source limits
(delegation limits) in canvas apps



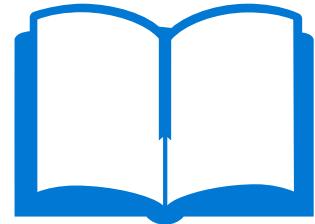
© Copyright Microsoft Corporation. All rights reserved.

180

Microsoft Learn module

Work with data source limits (delegation limits) in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/work-with-data-source-limits-powerapps-canvas-app>



© Copyright Microsoft Corporation. All rights reserved.

181

Delegation overview

Before you choose your data source in Power Apps, it's important to understand delegation.

Delegation in action

```
Filter(SharePointList,  
ProjectStatus = "Active")
```

When delegation isn't available

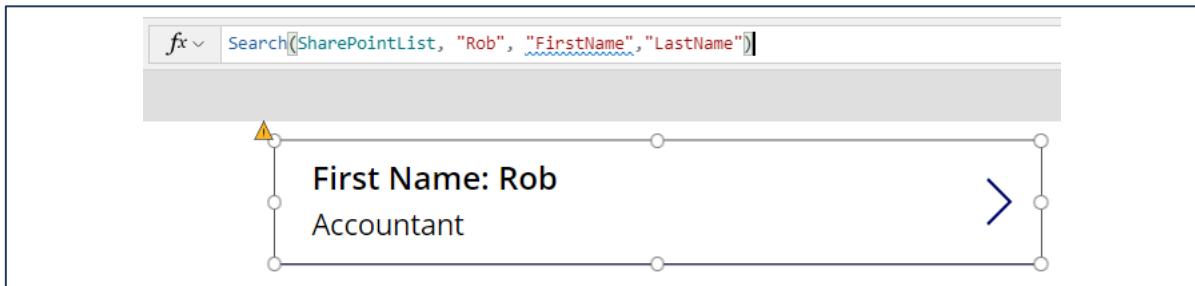
Consider delegation when choosing a data source

© Copyright Microsoft Corporation. All rights reserved.

182

Delegation warnings

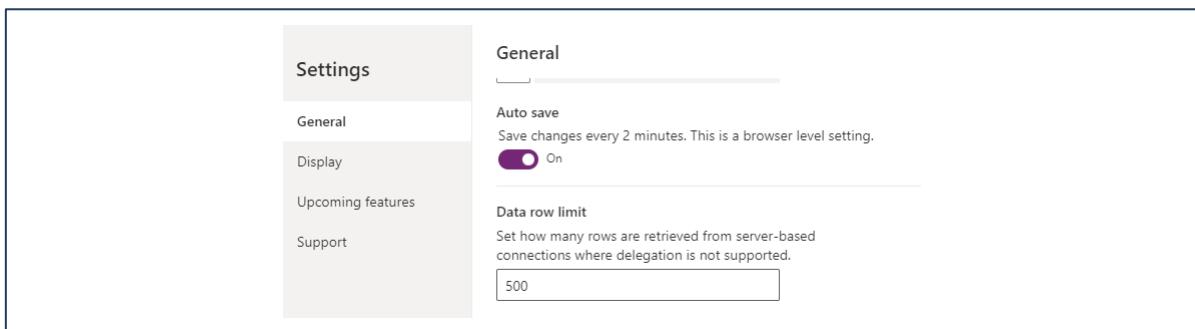
Anytime you use a non-delegable function, Power Apps underlines it with a blue line and displays a yellow warning triangle as shown below.



183

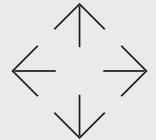
Changing the number of rows returned when delegation isn't available

When a formula cannot delegate to the data source, for any reason, then by default the canvas app retrieves the first 500 rows from that data source and the processes the formula locally



184

Module 6: Performance in canvas apps



© Copyright Microsoft Corporation. All rights reserved.

185

Microsoft Learn module

Complete testing and performance checks in a Power Apps canvas app

<https://learn.microsoft.com/training/modules/testing-performance-checks-powerapps>

Optimize app load time

<https://learn.microsoft.com/training/modules/optimize-app-load-time>

Use Monitor to troubleshoot Power Apps

<https://learn.microsoft.com/training/modules/monitor-to-troubleshoot>

Use Power Apps Instrumentation with Application Insights

<https://learn.microsoft.com/training/modules/instrumentation-app-insights>



© Copyright Microsoft Corporation. All rights reserved.

186

The importance of thinking about performance

The most common app performance problems come from interactions with data sources

Storing data in the wrong data source

Too many lookups

Too many refreshes

© Copyright Microsoft Corporation. All rights reserved.

187

Other performance considerations

While data sources might be the largest bottlenecks, there are other easily overlooked changes you can make to get optimal performance.

- **Asset size** - When you are designing your app it is great to include company logos and other visual assets. When you add these assets to your app, make sure the assets are optimized for the size of your app. The higher the resolution of a file, the larger the file size, and the more resources it takes for your app to store and display the image. Use an image editing tool to resize your files to the size you need for your app.
- **Republish your app** - The Power Apps team is constantly updating Power Apps to bring new features and to increase performance. The only way that your app takes advantage of these advancements is for you to open the app and publish again. Your app will stay on the version that it was published on until you do this. So periodically revisiting your app to move to the latest version will provide you with the best possible performance.
- **Build focused apps** - Power Apps supports building apps with as many screens as you can imagine, but too many screens is not a good idea. You should build your apps focused on a specific audience and process. This allows you to optimize the user experience for one audience, makes building and troubleshooting the app easier, and reduces the size of the app. If you have one app for everything, consider breaking it into smaller apps by role.

© Copyright Microsoft Corporation. All rights reserved.

188

Improve performance with data sources

Use collections to cache data

```
Filter(DepartmentList, Status = "Active")
```

```
Collect(collectDepartmentList, Filter(DepartmentList, Status = "Active"))
```

Delegation also affects performance

© Copyright Microsoft Corporation. All rights reserved.

189

Use the Concurrent function to load multiple data sources

```
Collect(collectDepartmentList,  
Filter(DepartmentList, Status = "Active"));  
Collect(collectCompanyList, CompanyList);  
Collect(collectRegions, RegionList);
```

```
Concurrent(  
    Collect(collectDepartmentList,  
    Filter(DepartmentList, Status = "Active")),  
    Collect(collectCompanyList, CompanyList),  
    Collect(collectRegions, RegionList)  
)
```

© Copyright Microsoft Corporation. All rights reserved.

190

OnStart versus OnVisible

OnStart and OnVisible are part of your toolkit for building great apps, but from a performance point of view, they can have a major impact.

- OnStart - This is an app-level property. Formulas in this property are run once, when the app starts, and never again. All of the formulas must process before the app opens. This is often used to initialize data that you will need throughout the app.
- OnVisible - This is a per-screen property. Formulas in this property are run every time a user navigates to the screen. The screen will render before the formula is finished processing.

© Copyright Microsoft Corporation. All rights reserved.

191

Optimize app load time

Maker portal, includes the **Time to first screen** and **Time to first screen without connection setup** analytics.

Review app settings

Use a loading spinner image

Visible =

Delegation also affects performance

© Copyright Microsoft Corporation. All rights reserved.

192

Looking at the network activity of your app

Now that you have learned about testing from within the app you need to look at actual network calls and performance.

The screenshot shows the Microsoft Power Apps Monitor interface. On the left, there's a preview of a "Giant App" screen with a button labeled "Button" and a timer showing "00:00:01". On the right, a detailed network timeline shows numerous requests, mostly "application/json" type, with various status codes like OK, BadRequest, and InternalServerError. The timeline includes columns for Method, Result, Content-type, Requested, Time, Iteration, and Duration. To the right of the timeline, a summary of performance metrics is displayed:

Stalled:	0.9 ms
Resolving (DNS):	0.24 ms
Connecting (TCP):	40.05 ms
SSL:	19.85 ms
Sending:	0.23 ms
Waiting (TTFB):	1.41 s
Downloading:	100.85 ms

193

Monitor

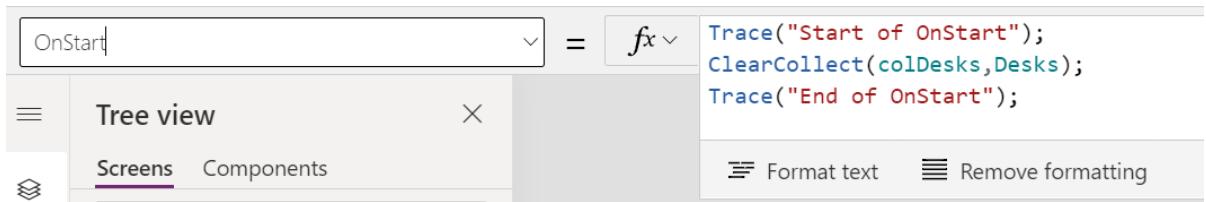
Monitor is a tool that you can launch from Microsoft Power Apps Studio to help you troubleshoot problems and improve the quality of your apps

The screenshot shows the Microsoft Power Apps Monitor tool window. It displays a log of network events for a published app session. The log table has columns for ID, Time, Category, Operation, Result, Status, Duration, Data source, Control, Property, Response, and Progress. The log entries show various actions such as ScreenLoad, UserAction, Network, and SetProperty, with details like success rates, duration, and data source. A progress bar at the bottom indicates the log is 32 items connected.

194

Trace

Add custom trace messages for Monitor



195

Learning Path 4 practice labs

Power Apps

Lab 4: Advanced canvas app techniques



© Copyright Microsoft Corporation. All rights reserved.

196



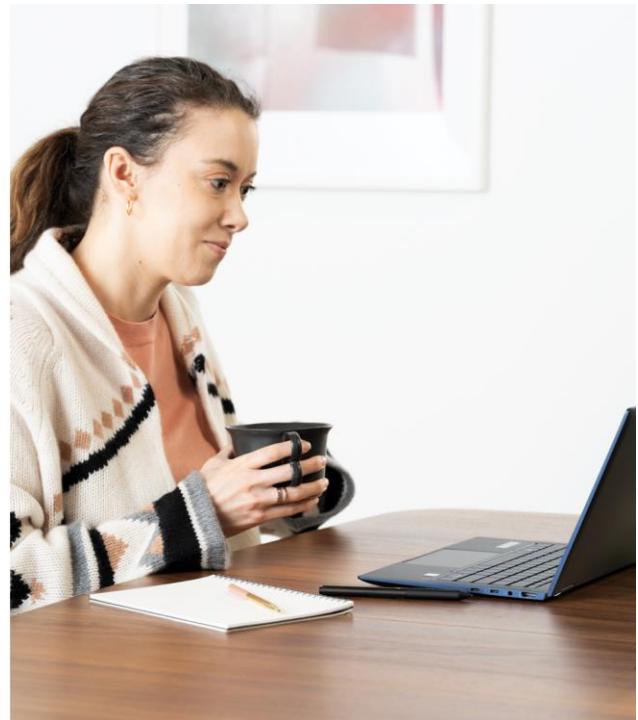
© Copyright Microsoft Corporation. All rights reserved.

197



Automate a business process using Power Automate

© Copyright Microsoft Corporation. All rights reserved.



198

Module 1: Get started with Power Automate



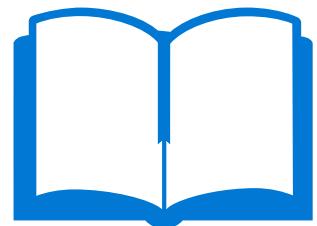
© Copyright Microsoft Corporation. All rights reserved.

199

Microsoft Learn module

Get started with Power Automate

<https://learn.microsoft.com/training/modules/get-started-flows>

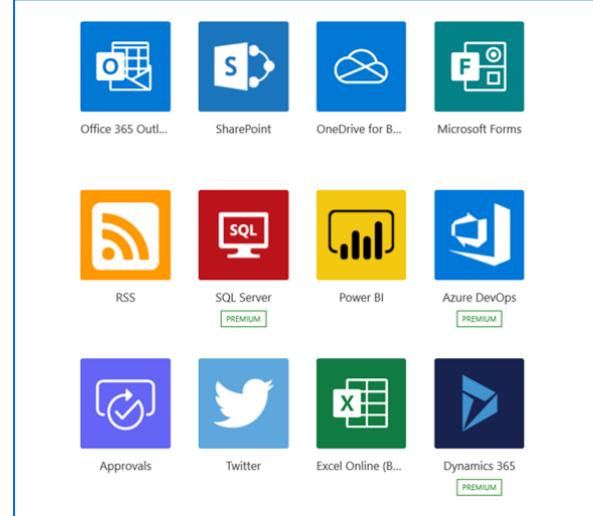


© Copyright Microsoft Corporation. All rights reserved.

200

What is Power Automate?

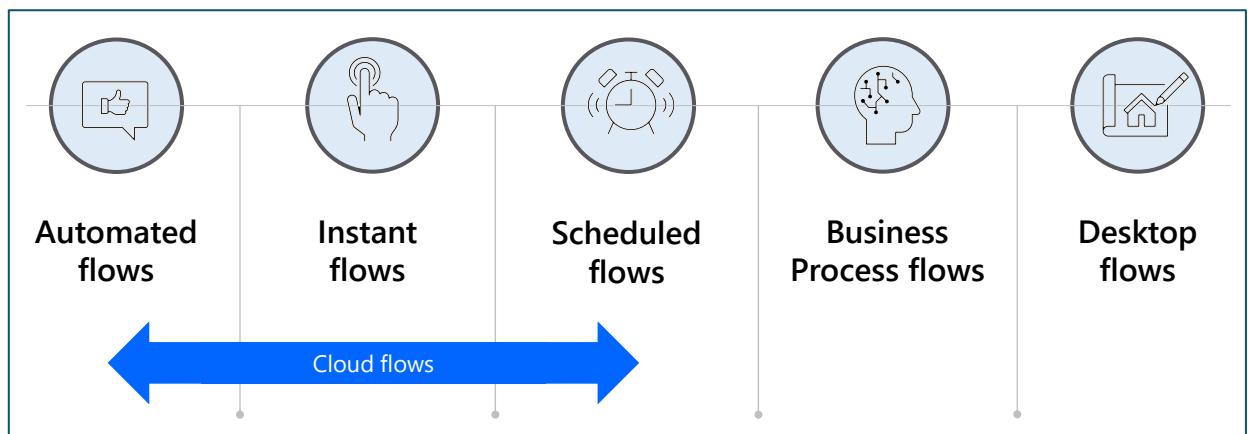
Power Automate is an online workflow service that automates actions across the most common apps and services



© Copyright Microsoft Corporation. All rights reserved.

201

Types of Power Automate flows



© Copyright Microsoft Corporation. All rights reserved.

202

A brief tour of Power Automate

Flows can be created from:
make.powerautomate.com

The screenshot shows the Microsoft Power Automate web interface. On the left is a navigation sidebar with options like Home, Action items, My flows, Create, Templates, Connectors, Data, Monitor, AI Builder, Process advisor, Solutions, and Learn. The main area features a banner with the text "Take care of what's important. Automate the rest." and "Create automated workflows with Microsoft Power Automate." Below the banner is a search bar labeled "Search for a template by app, task, or industry". A section titled "Start from a template" includes tabs for "Top picks", "Remote work", "Email", "Notifications", "Save to cloud", and "Approval". Under "Top picks", there are three cards: "Notify a channel when the status of a task in Planner updates" (Automated, 20236), "Save a message to OneNote" (Instant, 16860), and "Welcome new member to a team and share their bio" (Automated, 5464). At the bottom of the page is a copyright notice: "© Copyright Microsoft Corporation. All rights reserved."

203

Key concepts

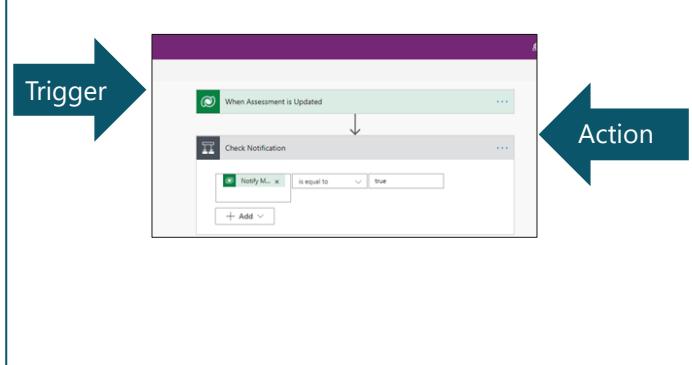
Every flow has two main parts:

A Trigger

The starting action of the flow such as new email arriving in your inbox or a new item being added to a SharePoint list

One or more Actions

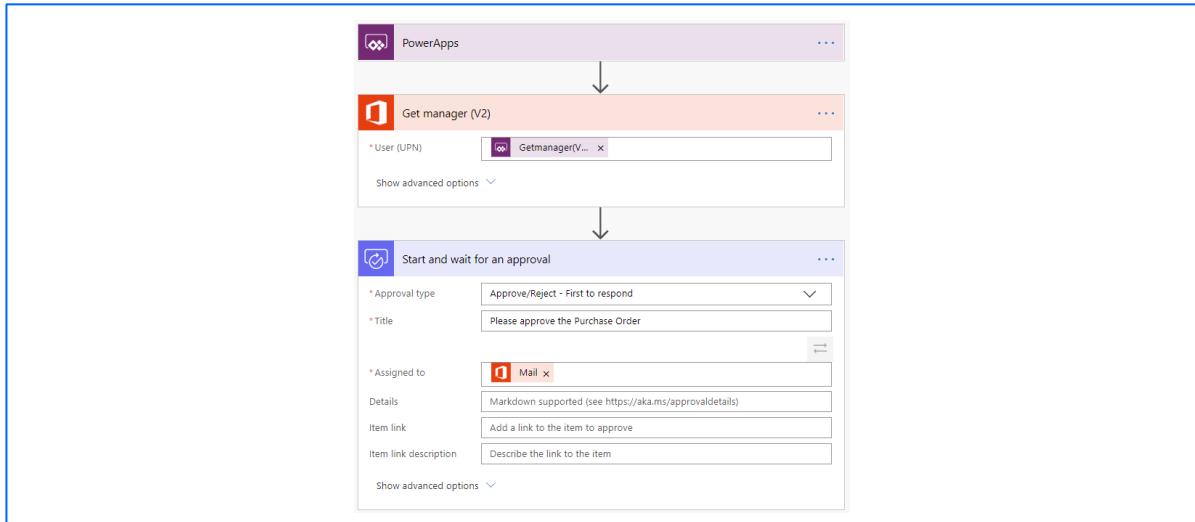
What you want to happen when a trigger is invoked such as start the action of creating a new file on OneDrive for Business



© Copyright Microsoft Corporation. All rights reserved.

204

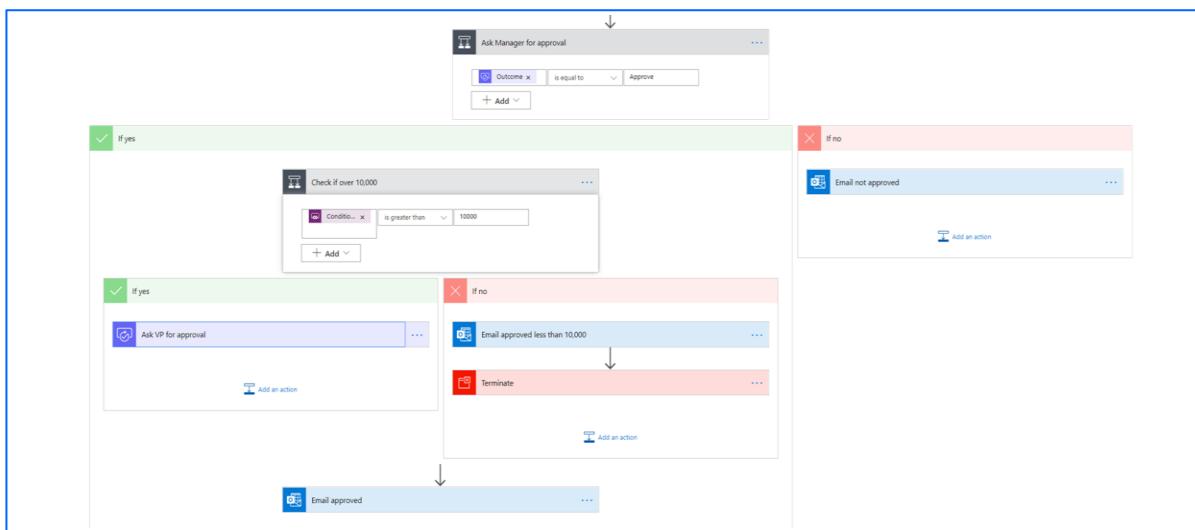
Example: Purchase order approval



© Copyright Microsoft Corporation. All rights reserved.

205

Example: Purchase order approval



© Copyright Microsoft Corporation. All rights reserved.

206

Flow building fundamentals

Dynamic data

Variables

Looping

Error Handling

Conditions

Expressions

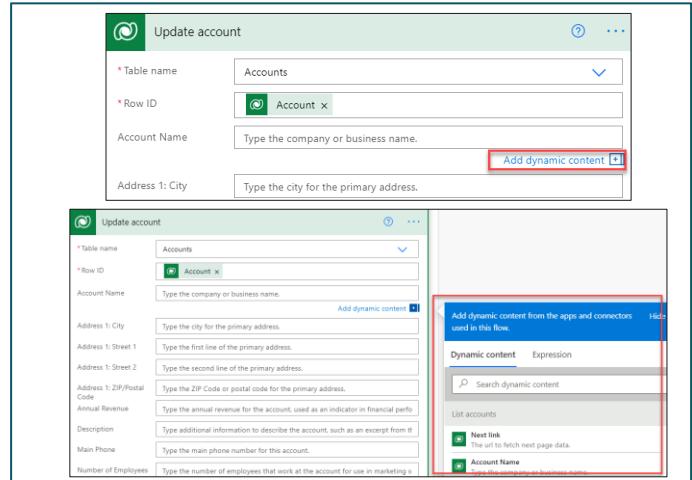
© Copyright Microsoft Corporation. All rights reserved.

207

Dynamic data

Use dynamic content panel to reference data from prior steps in a flow and dynamically bind it to current properties

Dynamic content panel shown when clicking in a property value on a step flow



© Copyright Microsoft Corporation. All rights reserved.

208

Conditions and switches

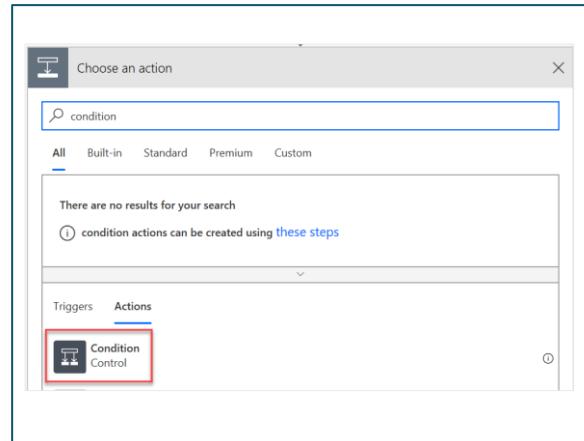
Conditions and switches offer ways to introduce conditional branching into a flow

Using conditions

Search actions on “Condition” and select

Using switches

Searching on “Switch” and add to flow



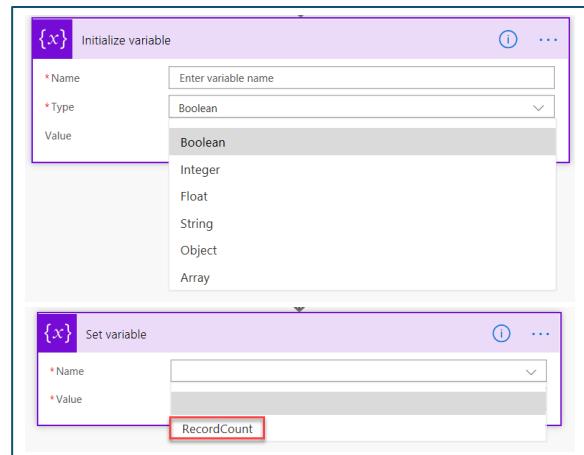
© Copyright Microsoft Corporation. All rights reserved.

209

Variables

First initialized, then set, incremented or decremented depending on their data type

Can be used in condition criteria to hold a value you plan to use later



© Copyright Microsoft Corporation. All rights reserved.

210

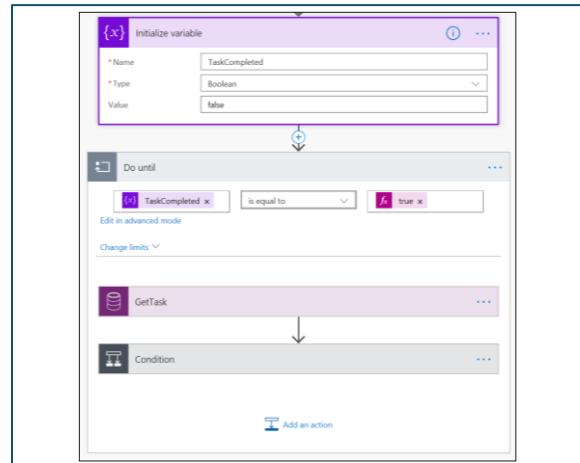
Looping

Apply for each

Executes for each item in an array

Do until

Executes until a condition is met



© Copyright Microsoft Corporation. All rights reserved.

211

Error handling

Goal: Recover work already completed by the flow before the error occurred

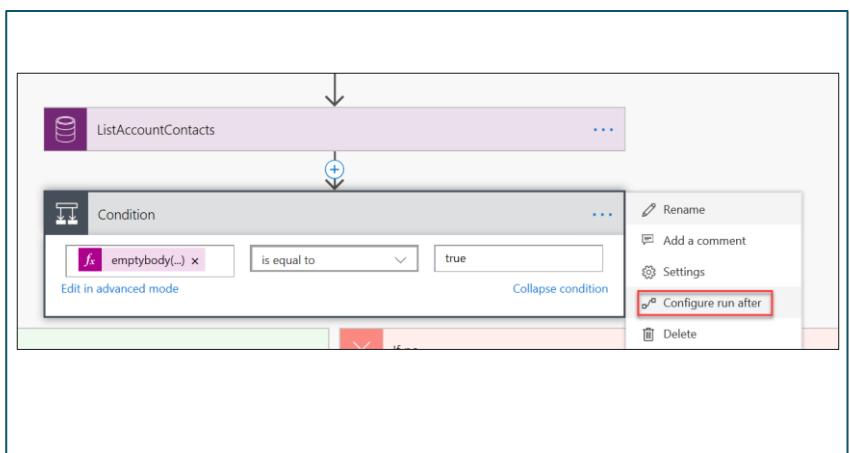
By default, flow will keep running through its step as long as prior actions have been successful

Errors:

Failed

Skipped

Timed out



© Copyright Microsoft Corporation. All rights reserved.

212

Administering flows

You can create a flow and perform administrative tasks in a browser or, if you download the Power Automate mobile app, on your phone:

Turn flows on or off from wherever you are

See when a flow has failed

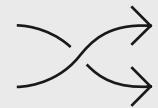
Review detailed run history reports

View and filter runs by notification type

© Copyright Microsoft Corporation. All rights reserved.

213

Module 2: Introduction to expressions in Power Automate



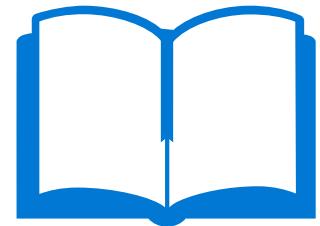
© Copyright Microsoft Corporation. All rights reserved.

214

Microsoft Learn module

Introduction to expressions in Power Automate

<https://learn.microsoft.com/training/modules/introduction-expressions>



© Copyright Microsoft Corporation. All rights reserved.

215

Expressions

The screenshot shows a Microsoft Power Automate flow editor. At the top, there's a 'Manually trigger a flow' step. Below it is a 'Delay' step, which has a 'Count' field set to 'Specify the count of unit to delay' and a 'Unit' field set to 'Minute'. A red box highlights the 'Count' field. Below the 'Delay' step is a 'Send a push notification' step. At the bottom of the flow are 'New step' and 'Save' buttons. To the right of the flow, a large window titled 'Expression' shows a list of functions. The 'add' function is selected and highlighted in blue. Other visible functions include 'OK', 'String functions', 'Collection', 'Logical functions', and 'Date and time functions'. Each function has a brief description and a small icon next to it.

© Copyright Microsoft Corporation. All rights reserved.

216

Types of functions

Functions are grouped into 10 different categories like math and logic.

Add an expression to do basic things like access, convert, and compare values. [Learn more about dynamic content](#)

Dynamic content Expression

OK

String functions [See more](#)

- fx concat(text_1, text_2, ...)** Combines any number of strings together
- fx contains(collection, value)** Returns true if a dictionary contains a key, if an array cont...
- fx length(collection)** Returns the number of elements in an array or string

Logical functions [See more](#)

- fx if(expression, valueIfTrue, valueIfFalse)** Returns a specified value based on whether the expressio...
- fx equals(object1, object2)** Returns true if two values are equal
- fx and(expression1, expression2)** Returns true if both parameters are true

Dynamic content **Expression**

fx

OK

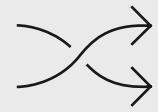
String functions [See less](#)

- fx concat(text_1, text_2, ...)** Combines any number of strings together
- fx substring(text, startIndex, length?)** Returns a subset of characters from a string
- fx replace(text, oldText, newText)** Replaces a string with a given string
- fx guid()** Generates a globally unique string (GUID)
- fx toLower(text)** Converts a string to lowercase using the casing rules of th...
- fx toUpper(text)** Converts a string to uppercase using the casing rules of th...

© Copyright Microsoft Corporation. All rights reserved.

217

Module 3: Use Dataverse triggers and actions in Power Automate



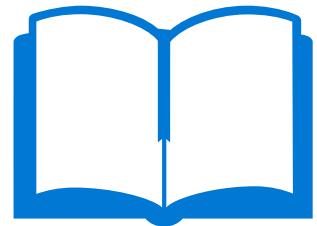
© Copyright Microsoft Corporation. All rights reserved.

218

Microsoft Learn module

Use Dataverse triggers and actions in Power Automate

<https://learn.microsoft.com/training/modules/dataverse-triggers-actions>



© Copyright Microsoft Corporation. All rights reserved.

219

Microsoft Dataverse Connectors



- Use when you need to integrate across environments
- Need to use for When a row is selected



- Use when talking to Microsoft Dataverse in the same environment
- This is the preferred connector to use and will replace the other two connectors

© Copyright Microsoft Corporation. All rights reserved.

220



Microsoft Dataverse Triggers and Actions

Triggers

- When a flow step is run from a business process flow PREMIUM
Microsoft Dataverse
- When a row is added, modified or deleted PREMIUM
Microsoft Dataverse
- When an action is performed (preview) PREMIUM
Microsoft Dataverse

Actions

- Add a new row PREMIUM
Microsoft Dataverse
- Delete a row PREMIUM
Microsoft Dataverse
- Download a file or an image PREMIUM
Microsoft Dataverse
- Get a row by ID PREMIUM
Microsoft Dataverse
- List rows PREMIUM
Microsoft Dataverse
- Perform a bound action PREMIUM
Microsoft Dataverse
- Perform an unbound action PREMIUM
Microsoft Dataverse
- Relate rows PREMIUM
Microsoft Dataverse
- Search rows (preview) PREMIUM
Microsoft Dataverse
- Unrelate rows PREMIUM
Microsoft Dataverse
- Update a row PREMIUM
Microsoft Dataverse
- Upload a file or an image PREMIUM
Microsoft Dataverse
- Perform a changeset request PREMIUM
Microsoft Dataverse

© Copyright Microsoft Corporation. All rights reserved.

221



Dataverse data change trigger

Specify for trigger

- Change type
- Table name
- Scope

When a row is added, modified or deleted

* Change type:

* Table name:

* Scope:

Show advanced options

When a row is added, modified or deleted

Show options

* Change type:

Added
Added or Deleted
Added or Modified
Added or Modified or Deleted
Deleted
Modified
Modified or Deleted
Enter custom value

© Copyright Microsoft Corporation. All rights reserved.

222

Scope on triggers

Organization

Trigger for all rows user has read access to

Business unit

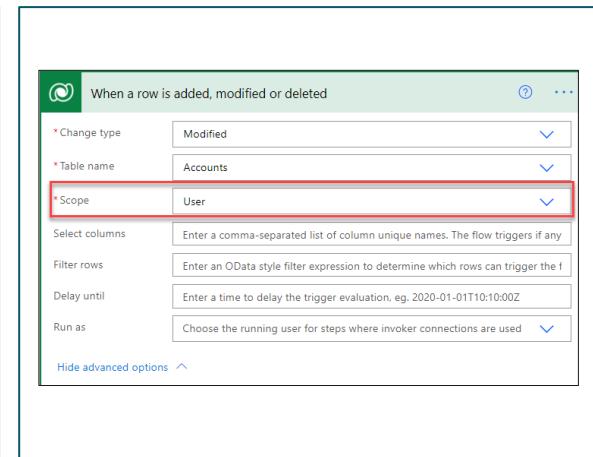
Only trigger for all rows user has read access to and owned by a user in same business unit

Business unit + child business unit

Only trigger for all rows user has read access to and owned by a user in same business unit and hierarchy of child business units

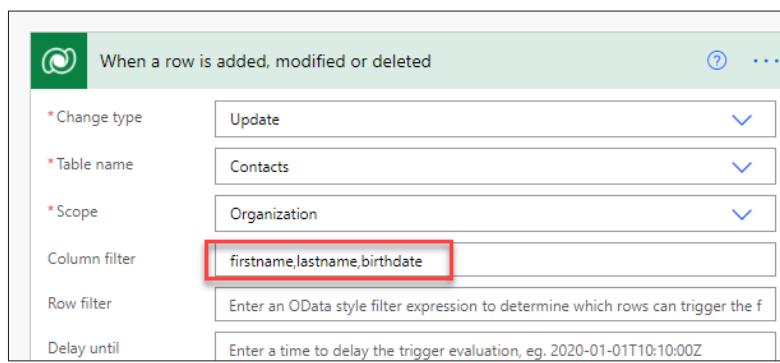
User

Only trigger for modifications to rows they own



223

Filtering Update Triggers



224

© Copyright Microsoft Corporation. All rights reserved.



Filtering Update Triggers

The screenshot shows the configuration of a Microsoft Flow trigger. The trigger type is "When a row is added, modified or deleted". The settings include:

- * Change type: Modified
- * Table name: Contacts
- * Scope: Organization
- Select columns: Enter a comma-separated list of column unique names. The flow tries to guess them based on the table.
- Filter rows: statuscode eq 1 (highlighted with a red box)
- Delay until: Enter a time to delay the trigger evaluation, eg. 2020-01-01T10:10:00Z
- Run as: Select an item

© Copyright Microsoft Corporation. All rights reserved.

225

Querying data

Get a row by ID action

Retrieve a single row

List rows action

Retrieve an array of rows

- OData filter
- FetchXML query

The screenshot shows the configuration of a Microsoft Flow action. The action type is "Get primary contact". The settings include:

- * Table name: Contacts
- * Row ID: Primary Contact... (highlighted with a blue box)

Dynamic content (Expression):

- Search bar: primary
- When a row is added, modified or deleted:
 - Primary Contact (Type): Choose the primary contact for the account to provide query results.
 - Primary Contact (Value): Choose the primary contact for the account to provide query results.
 - Address 1: Shows the complete primary address.

© Copyright Microsoft Corporation. All rights reserved.

226

Creating and updating data

Add a new row

Provide minimum required columns to create a row

Update a row

Provide item identifier

Set column to null to clear

The screenshot shows a form with several fields. The 'Primary Contact' field is highlighted with a red box and contains the value 'null'. To its right, a modal window titled 'Expression' is open, also containing the value 'null'. Both the field and the modal are highlighted with red boxes.

© Copyright Microsoft Corporation. All rights reserved.

227

Setting lookup column values

Uses OData syntax

The screenshot shows a form with a 'Company Name (Accounts)' field. A red box highlights the input field, which contains the value 'Account'. To the right of the field, a modal window titled 'Expression' is open, showing the value 'Account'. Both the field and the modal are highlighted with red boxes.

Incorrect Method: On the left, there is a large red 'X' icon next to a form field with a red box around the 'Account' button.

Correct Method: On the left, there is a large black checkmark icon next to a form field with a red box around the 'Account' button. The 'Expression' modal window is also shown on the right, listing 'Create Account' and other options.

© Copyright Microsoft Corporation. All rights reserved.

228



Run flows on selected records trigger

The screenshot shows the Microsoft Flow Designer interface. A flow is being created with the following steps:

- When a record is selected** (trigger, highlighted with a red box)
- Get opportunity row**
- Condition**

Below the flow editor, there is a configuration panel for the trigger:

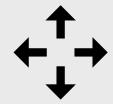
- * Environment:** Default
- * Table Name:** Accounts
- Choose the type of user input:**
 - Text
 - Yes/No
 - File
 - Email
 - Number
 - Date

On the right side of the screen, a screenshot of the Microsoft Dynamics 365 interface shows a context menu for a record. The 'Flow' option is selected, and a sub-menu is open with the 'Run' option highlighted by a red arrow. A blue curved arrow points from the 'Run' option in the menu to the 'Run' step in the Flow Designer.

© Copyright Microsoft Corporation. All rights reserved.

229

Module 4: Advanced features of cloud flows



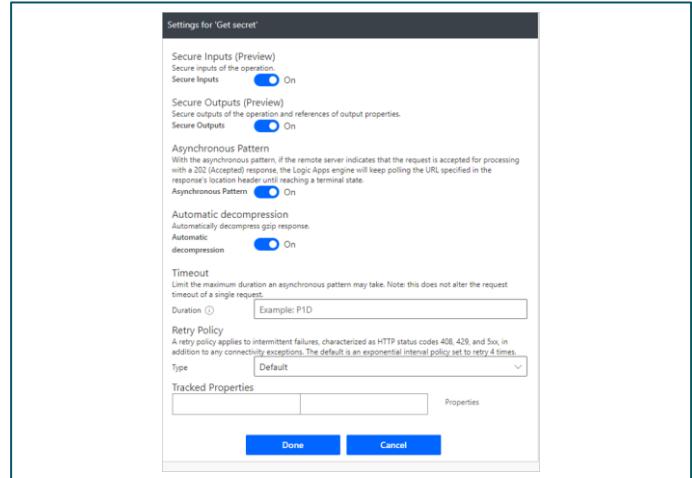
© Copyright Microsoft Corporation. All rights reserved.

230

Sensitive data in cloud flow steps

Secure Inputs

Secure Outputs



© Copyright Microsoft Corporation. All rights reserved.

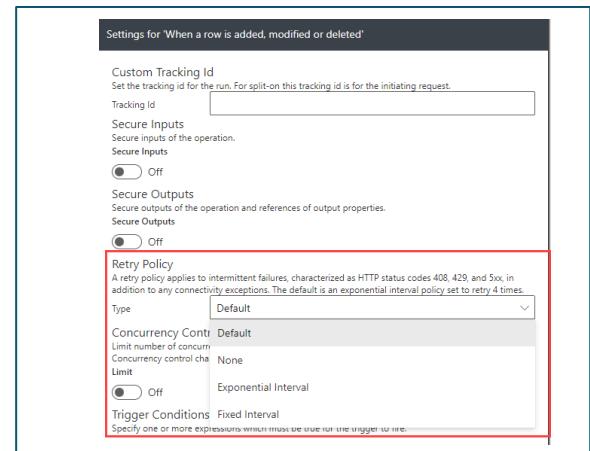
231

Retry policy and concurrency

Policy

- Default
- Exponential
- Fixed

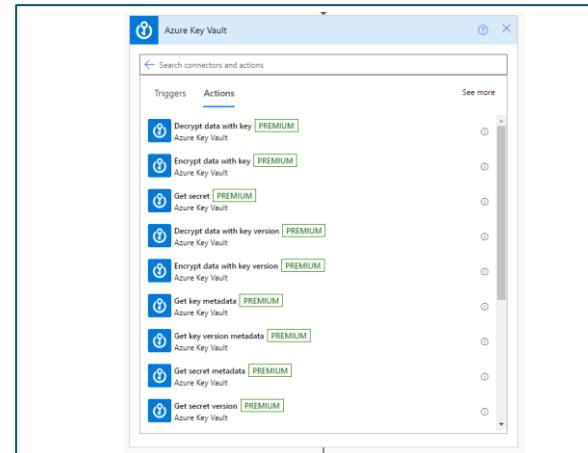
Concurrency control



© Copyright Microsoft Corporation. All rights reserved.

232

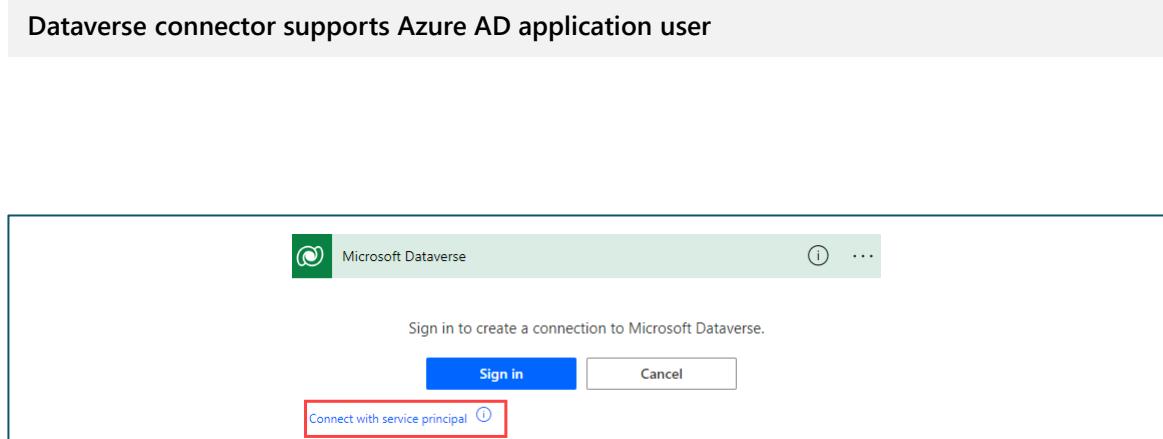
Azure Key Vault



© Copyright Microsoft Corporation. All rights reserved.

233

Azure Active Directory service principals



© Copyright Microsoft Corporation. All rights reserved.

234

Cloud flow step in a business process flow

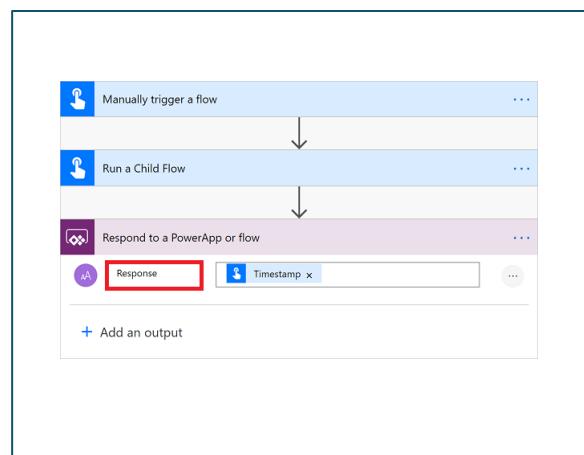
Business process flows	Cloud flow step
Guides a user through business process stages in a model-driven app	When a flow step is run from a business process flow trigger
Started when a row is created	Instant trigger
Each BPF has a table created automatically to track current stage	Flow runs on the BPF table

© Copyright Microsoft Corporation. All rights reserved.

235

Child flows

Call child flows from parent flow
<ul style="list-style-type: none"> Run a Child flow action
Child flow triggers
<ul style="list-style-type: none"> Power Apps Manually trigger a flow
Pass data back to parent flow
<ul style="list-style-type: none"> Respond to a Power Apps or flow action Response (from Request connector) action



© Copyright Microsoft Corporation. All rights reserved.

236

Cloud flows and solutions



Instant flows must be created in a solution



Parent and child flows must be in the same solution



Cloud flows used in Business process flows must be in a solution

© Copyright Microsoft Corporation. All rights reserved.

237

Calling a cloud flow from a Canvas app

Call cloud flows from canvas app

- <Flow Name>.Run(parameters)

Flow trigger

- Power Apps
- Use Ask in Power Apps to create parameters

Pass data back to app

- Respond to a Power Apps or flow action
- Use Set to store response in a variable

```
Set(varResult,
SendEmailFlow.Run(
EditForm3.LastSubmit.Email,
TextInput1.Text)
.result
);
```

© Copyright Microsoft Corporation. All rights reserved.

238

Learning Path 5 practice labs

Power Automate

Lab 5: Power Automate



© Copyright Microsoft Corporation. All rights reserved.

239



© Copyright Microsoft Corporation. All rights reserved.

240



Learning Path 6: Introduction to developing with Power Platform

© Copyright Microsoft Corporation. All rights reserved.



241

Module 1: Introduction to Power Platform developer resources



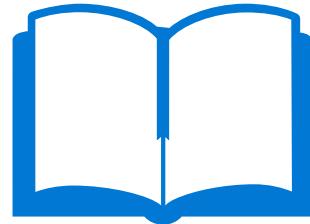
© Copyright Microsoft Corporation. All rights reserved.

242

Microsoft Learn module

Introduction to Microsoft Power Platform developer resources

<https://learn.microsoft.com/training/modules/introduction-developing-power-platform>



© Copyright Microsoft Corporation. All rights reserved.

243

When to write code

As a Power Platform developer, you need to understand the gaps that exist between what can be accomplished through configuration versus code. Sometimes, existing features might not provide the functionality needed to meet a requirement, and Dataverse provides various points where developers can extend the common functionality by using code.

To identify gaps where features don't already exist, it's crucial that all Power Platform developers familiarize themselves with the capabilities of Dataverse.

© Copyright Microsoft Corporation. All rights reserved.

244

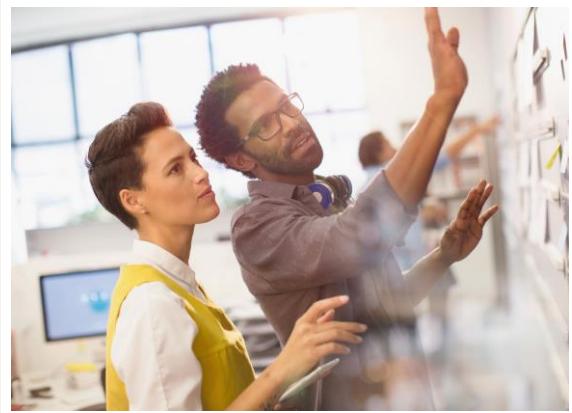
Types of extensibility

Extending the user experience of a model-driven app

Extending the user experience of a Canvas apps

Extending the Microsoft Dataverse platform

Extending by leveraging Azure services



© Copyright Microsoft Corporation. All rights reserved.

245

Explore the Power Platform Developer Guide

A wealth of assets is available that developers can review to better acquaint themselves on developing against the Dataverse framework.

Sample code

<https://github.com/microsoft/PowerApps-Samples>
<https://github.com/pnp/powerapps-samples>

Useful links

Microsoft Dataverse developer documentation
<https://learn.microsoft.com/power-apps/developer/data-platform>
 Model-driven apps developer <https://learn.microsoft.com/power-apps/developer/model-driven-apps>
 Power Apps component framework
<https://learn.microsoft.com/power-apps/developer/component-framework/overview>

© Copyright Microsoft Corporation. All rights reserved.

246

Module 2: Use developer tools to extend the Power Platform



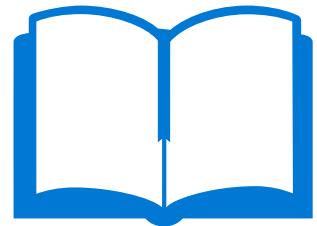
© Copyright Microsoft Corporation. All rights reserved.

247

Microsoft Learn module

Use developer tools to extend Microsoft Power Platform

<https://learn.microsoft.com/training/modules/use-developer-tools-extend>



© Copyright Microsoft Corporation. All rights reserved.

248

Introduction to developer Microsoft Power Platform tooling

Microsoft Power Apps CLI	Microsoft Power Platform connectors CLI	Application Lifecycle Management (ALM) tools	Microsoft NuGet tools for Dataverse	Community tools
---------------------------------	--	---	--	------------------------

© Copyright Microsoft Corporation. All rights reserved.

249

Microsoft NuGet tools

Microsoft provides a variety of tools to help support development against Dataverse, which are available through [NuGet packages](#) and the Power Platform CLI .

Code Generation Tool	Configuration Migration Tool	Package Deployer	Plugin Registration Tool	Solution Packager Tool
-----------------------------	-------------------------------------	-------------------------	---------------------------------	-------------------------------

© Copyright Microsoft Corporation. All rights reserved.

250

Learning Path 6 practice labs

Introduction to developing with the Power Platform

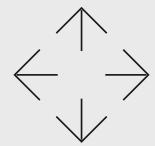
Lab 6: Power Platform tools



© Copyright Microsoft Corporation. All rights reserved.

251

Module 3: Introduction to extending the Microsoft Power Platform



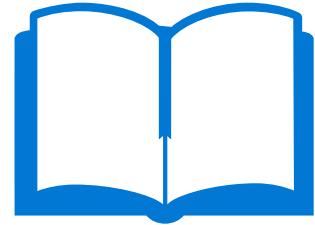
© Copyright Microsoft Corporation. All rights reserved.

252

Microsoft Learn module

Introduction to extending Microsoft Power Platform

<https://learn.microsoft.com/training/modules/introduction-power-platform-extensibility-model>



© Copyright Microsoft Corporation. All rights reserved.

253

Power Apps code component framework (PCF)

Code components are implemented using HTML, CSS, and TypeScript.

```

<manifest>
  <control name="FirstControl" constructor="FirstControl" version="0.0.1" display-name-key="FirstControl" description-key="FirstControl_desc">
    <!-- property node identifies a specific, configurable piece of data that the control expects from CPS -->
    <property name="companyName" display-name-key="Property_Display_Key" description-key="Property_Desc_Key"
      of-type="SingleLineText" usage="bound" required="true" />
  </control>
</manifest>

```

254

Client scripting

Client scripting allows you to use JavaScript in model-driven apps to implement business logic



Form load



Data in a column changes



Form is saved

© Copyright Microsoft Corporation. All rights reserved.

255

Using the Dataverse APIs

Dataverse provides two styles of APIs that developers can use to interact with data



Web API - The Web API is available at an OData v4 RESTful endpoint. Use this for any programming language that supports HTTP requests and authentication using OAuth 2.0.



Organization Service - The Organization Service is a .NET Framework SDK with .NET assemblies provided by Microsoft along with typed class generators for table classes. When you access data from a plugin this is how you access data. The service is instantiated and available to the plugin code without the need to authenticate.

© Copyright Microsoft Corporation. All rights reserved.

256

Building plugins



Plugins are .NET classes that implement a **IPlugin** interface provided by the Dataverse SDK assemblies



Implement business logic within the event pipeline

© Copyright Microsoft Corporation. All rights reserved.

257

Custom APIs



Operations in Dataverse are defined as messages



Custom APIs create new messages in the event pipeline



Write a plug-in to perform the logic for the custom API

© Copyright Microsoft Corporation. All rights reserved.

258

Determine when to configure or when to code

As a developer, you should approach apps on Microsoft Power Platform from the perspective that writing code for achieving desired business application functionality should be considered as an exception to no-code and low-code approaches

259

Business rules vs. client script

The advantage of business rules is that they are easy to understand and implement for a non-developer, and they can be included as part of a Dataverse solution for deployment in production.

Workflows/Power Automate flows vs. plugins

Dataverse offers various methods to implement logic to respond to events in the system, in particular to the data changes such as create, update, delete of the data rows.

Capability	Power Automate cloud flow	Dataverse classic workflow	Plugin
Synchronous or Asynchronous	Asynchronous	Either (Power Automate flows are recommended instead of asynchronous workflows)	Either
Access External Data	Yes, using connectors	No	Yes, using APIs, some security restrictions
Maintenance	Makers	Business Users	Developers
Can Run As	Current user or flow owner	Current user or workflow owner	Any licensed user, system, or current user
Can Run On Demand	Yes	Yes	No
Can Nest Child Processes	Yes	Yes	Yes
Execution Stage	After	Before/After	Before/After
Triggers	Create, Column Change, Delete, On Demand, Scheduled	Create, Column Change, Delete, On Demand	Create, Column Change, Delete, other messages including custom

© Copyright Microsoft Corporation. All rights reserved.

261

Module 4: Work with Dataverse APIs



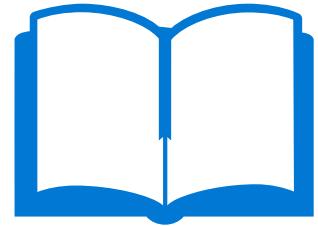
© Copyright Microsoft Corporation. All rights reserved.

262

Microsoft Learn module

Work with Dataverse Web API

<https://learn.microsoft.com/training/modules/common-data-service-web-api>



© Copyright Microsoft Corporation. All rights reserved.

263

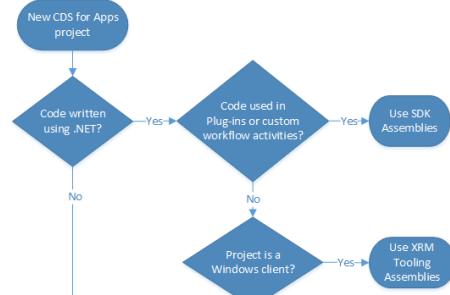
Web API vs. the Organization Service

Web API

- REST
- OData 4.0 compliant

Organization Service

- .NET Required for Plug-ins
- XRM Tooling for Windows client applications



© Copyright Microsoft Corporation. All rights reserved.

264

Organization service



NET Framework (CRM SDK) or .NET Core (Dataverse SDK)



Add SDK assemblies using NuGet



IOrganizationService Interface



Early vs Late binding

© Copyright Microsoft Corporation. All rights reserved.

265

IOrganizationService Methods

The IOrganizationService interface has 8 methods:

- Create
- Retrieve
- Update
- Delete
- RetrieveMultiple
- Associate
- Disassociate
- **Execute**

© Copyright Microsoft Corporation. All rights reserved.

266

Early bound vs Late bound

Using the Organization service

Late bound	Early bound
------------	-------------

Entity class

Generate set of classes for each table

Refer to tables and columns as strings

Intellisense in Visual Studio

Run time validation

Compile time validation

© Copyright Microsoft Corporation. All rights reserved.

267

Create a row

```
var account = new Entity("account");
account["name"] = "Contoso";
account["creditonhold"] = false;
account["numberofemployees"] = 500;
account["revenue"] = new Money(new decimal(5000000.00));
account["accountcategorycode"] = new OptionSetValue(1); //Preferred customer
Guid accountid = svc.Create(account);
```

© Copyright Microsoft Corporation. All rights reserved.

268

Retrieve a row

```
Entity accountEntity = svc.Retrieve(  
    "account",  
    accountid,  
    new ColumnSet("name", "accountnumber"));
```

© Copyright Microsoft Corporation. All rights reserved.

269

Querying with the Organization service

Different methods for querying data

- QueryExpression
- QueryByAttribute
- FetchExpression (with FetchXML)
- LINQ

© Copyright Microsoft Corporation. All rights reserved.

270

Batching requests

ExecuteMultipleRequest

- ContinueOnError = true/false
- ReturnResponses = true/false

Maximum batch size = 1000 requests

```
ExecuteMultipleRequest requestWithResults = new ExecuteMultipleRequest()
{
    Settings = new ExecuteMultipleSettings()
    {
        ContinueOnError = false,
        ReturnResponses = true
    },
    Requests = new OrganizationRequestCollection()
};

EntityCollection input = GetCollectionOfEntitiesToDelete();
foreach (var entity in input.Entities)
{
    CreateRequest createRequest = new CreateRequest { Target = entity };
    requestWithResults.Requests.Add(createRequest);
}

ExecuteMultipleResponse responseWithResults =
    (ExecuteMultipleResponse)service.Execute(requestWithResults);
```

© Copyright Microsoft Corporation. All rights reserved.

271

Transactions

ExecuteTransactionRequest

- ReturnResponses = true/false

Maximum batch size = 1000 requests

```
ExecuteTransactionRequest requestWithResults = new ExecuteTransactionRequest()
{
    Settings = new ExecuteMultipleSettings()
    {
        ReturnResponses = true
    },
    Requests = new OrganizationRequestCollection()
};

EntityCollection input = GetCollectionOfEntitiesToDelete();
foreach (var entity in input.Entities)
{
    CreateRequest service.Execute = new CreateRequest { Target = entity };
    requestWithResults.Requests.Add(service.Execute);
}

ExecuteTransactionResponse responseWithResults =
    (ExecuteTransactionResponse)service.Execute(requestWithResults);
```

© Copyright Microsoft Corporation. All rights reserved.

272

Concurrency

Optimistic Concurrency

Use UpdateRequest

Set ConcurrencyBehavior

- IfRowVersionMatches
- AlwaysOverwrite
- Default

```

Id = new Guid("a976763a-ba1c-e811-a954-000d3af451d6");
Entity retrievedAccount = svc.Retrieve("account", Id, Columnset(true));
var account = new Account();
account.Id = retrievedAccount.Id;
account.RowVersion = retrievedAccount.RowVersion;
account.CreditOnHold = true;
var request = new UpdateRequest()
{
    Target = account,
    ConcurrencyBehavior = ConcurrencyBehavior.IfRowVersionMatches
};
try
{
    svc.Execute(request);
}

```

© Copyright Microsoft Corporation. All rights reserved.

273

API Limits

Service Protection limits

Evaluated over a 5-minute rolling window

Combination of number of requests, execution time, and number of concurrent requests

Handle 429 error (Too many requests)

API limits

Evaluated over 24 hours

Based on license

Can purchase more API calls

© Copyright Microsoft Corporation. All rights reserved.

274

Dataverse Web API



OData v4 RESTful endpoint



Uses OAuth 2.0 for authentication



Microsoft provides MSAL library for OAuth 2.0 authentication



Verbs such as POST, GET, and PATCH

© Copyright Microsoft Corporation. All rights reserved.

275

Authentication vs. authorization

Authentication is the process or action of verifying the identity of a user or process.

Microsoft's solution to this verification process is Azure Active Directory (Azure AD). Azure AD supports a number of options to verify the identity of a user or process.

Abstracting your identity provider allows for a good separation of concerns because managing usernames and passwords can be a difficult (and risky) process.

Authorization is the process or action of verifying whether an authenticated user is authorized to access the resources that are being provided.

Presently, authorization to Dataverse is at the tenant level, and permissions are delegated to the application based on the current signed-in user.

Therefore, you won't use OAuth to govern app-level security, which is instead handled by means of the Power Apps admin center.

© Copyright Microsoft Corporation. All rights reserved.

276

Register Dataverse apps with Azure AD

To successfully connect to your Dataverse, you must first register an app with Azure Active Directory, which can be completed in the Azure portal.

The screenshot shows the Azure Active Directory - App registrations interface. On the left, there's a sidebar with various management options like Overview, Getting started, Diagnose and solve problems, Manage (Users, Groups, etc.), and App registrations (which is highlighted with a red box and labeled 1). At the top, there's a search bar, a '+ New registration' button (highlighted with a red box and labeled 2), and tabs for All applications and Owned applications. Below the tabs is a search bar with placeholder text 'Start typing a name or Application ID to filter these results'. The main area lists one application: 'MS Learn Sample App' (Display name), with columns for Application (client) ID, Created On (11/9/2019), and Certificates & secrets (Current).

© Copyright Microsoft Corporation. All rights reserved.

277

Register Dataverse apps with Azure AD

Specify the name of your app and what type of account access you need.

If you're registering a web app, specify a Redirect URI by going to the Authentication section, setting the type to Web, and then entering a redirect URI.

The screenshot shows the 'Register an application' form. It has three main sections: 'Name' (with a placeholder 'MS Learn Sample App'), 'Supported account types' (with a radio button selected for 'Accounts in any organizational directory (Any Azure AD directory - Multitenant)'), and 'Redirect URI (optional)' (containing 'Web https://callbackurl'). There are also 'Help me choose...' and 'Next Step' buttons at the bottom.

© Copyright Microsoft Corporation. All rights reserved.

278

Azure Active Directory App manifest

The screenshot shows the 'MS Learn Sample App - Manifest' editor. The left sidebar lists navigation options: Overview, Quickstart, Manage, Branding, Authentication, Certificates & secrets, Token configuration (preview), API permissions, Expose an API, Owners, Roles and administrators (Preview), and Manifest (marked with a red circle 1). The main area displays the app's manifest JSON. Several sections of the JSON are highlighted with red boxes and numbered circles: section 2 highlights the 'allowPublicClient: true' setting under 'authProperties'; section 3 highlights the 'oauth2AllowIdTokenImplicitFlow: true' setting under 'oauth2Permissions'; and section 4 highlights the 'oauth2AllowImplicitFlow: true' setting under 'oauth2Permissions'.

```

1
2 "id": "ada72c66-ad42-4ba9-92fd-3a6307d8e283",
3 "acceptMappedClaims": null,
4 "accessTokenAcceptedVersion": null,
5 "addIns": [],
6 "allowPublicClient": true, ②
7 "appId": "44afdf11c-c146-4128-b9ce-f00162044d75",
8 "appRoles": [],
9 "auth2AllowIdTokenImplicitFlow": false,
10 "createdDateTime": "2019-11-09T22:32:23Z",
11 "groupMembershipClaims": null,
12 "identifierUris": [],
13 "informationalUrls": {
14     "termsOfService": null,
15     "support": null,
16     "privacy": null,
17     "marketing": null
18 },
19 "keyCredentials": [],
20 "knownClientApplications": [],
21 "logoutUrl": null,
22 "logoutUrl1": null,
23 "name": "MS Learn Sample App", ③
24 "oauth2AllowIdTokenImplicitFlow": true, ④
25 "oauth2AllowImplicitFlow: true, ④
26 "oauth2Permissions": [],
27 "oauth2RequirePostResponse": false,
28 "optionalClaims": null,
29 "orgRestrictions": [],
30 "parentalControlSettings": {
31     "countriesBlockedForMinors": [],
32     "legalAgeGroupRule": "Allow"
33 },
34

```

© Copyright Microsoft Corporation. All rights reserved.

279

API permissions

The screenshot shows the 'Request API permissions' interface. The top bar includes 'Select an API' and tabs for 'Microsoft APIs' (selected), 'APIs my organization uses', and 'My APIs'. Below this, a section titled 'Commonly used Microsoft APIs' lists several services: Microsoft Graph, Azure Data Catalog, Azure DevOps, Azure Rights Management Services, Azure Service Management, Data Export Service for Microsoft Dynamics 365, Dynamics 365 Business Central, Dynamics CRM (highlighted with a red box), Flow Service, and Intune. The Dynamics CRM row is specifically highlighted with a red box.

The screenshot shows the 'Request API permissions' dialog for Dynamics CRM. It lists 'Delegated permissions' (Your application needs to access the API as the signed-in user) and 'Application permissions' (Your application runs as a background service or daemon without a signed-in user). Under 'Permission', the 'user_impersonation' checkbox is checked, with the note 'Access Common Data Service as organization users'. The 'Admin Consent Required' checkbox is also present.

© Copyright Microsoft Corporation. All rights reserved.

280

Interact with Dataverse Web API by using Postman

The image shows two side-by-side screenshots. On the left is a screenshot of the Microsoft Dynamics 365 Settings page, specifically the 'Advanced settings' section which is highlighted with a red box. On the right is a screenshot of the 'Developer Resources' section of the Microsoft Dataverse documentation, showing links for 'Getting Started' and sections for 'Instance Web API', 'Instance Reference Information', and 'Connect your apps to the Dynamics 365 Discovery Service'. A red box highlights the 'Service Root URL' field containing the value 'https://c.*[REDACTED]*.api.crm.dynamics.com/api/data/v9.1/'.

© Copyright Microsoft Corporation. All rights reserved.

281

Use OData to query data

Create rows

```
OData
POST [Organization URI]/api/data/v9.2/accounts HTTP/1.1
Content-Type: application/json; charset=utf-8
OData-MaxVersion: 4.0
OData-Version: 4.0
Accept: application/json
```

```
{
    "name": "Sample Account",
    "creditonhold": false,
    "address1_latitude": 47.639583,
    "description": "This is the description of the sample account",
    "revenue": 5000000,
    "accountcategorycode": 1
}
```

282

Use OData to query data

Retrieve rows

OData

```
GET [Organization URI]/api/data/v9.2/accounts(00000000-0000-0000-0000-000000000001)
```

283

Use FetchXML to query data

FetchXML is a robust query language that was developed by Microsoft to enable complex operations to be performed against Dataverse data.

FetchXML can be used by both

- Dataverse Web API
- Organization Service

284

Example FetchXML query

XML

```
<fetch mapping='logical'>
  <entity name='account'>
    <attribute name='accountid'/>
    <attribute name='name'/>
  </entity>
</fetch>
```

XML

```
<fetch mapping='logical'>
  <entity name='account'>
    <attribute name='accountid'/>
    <attribute name='name'/>
    <filter type='and'>
      <condition
        attribute='address1_stateorprovince'
        operator='eq' value='WA' />
    </filter>
  </entity>
</fetch>
```

285

Learning Path 6 practice labs

Introduction to developing with the Power Platform

Lab 7: Power Platform APIs



© Copyright Microsoft Corporation. All rights reserved.

286



© Copyright Microsoft Corporation. All rights reserved.

287



Learning Path 7: Extending the model- driven app user experience

© Copyright Microsoft Corporation. All rights reserved.



288

Module 1: Performing common actions with client script



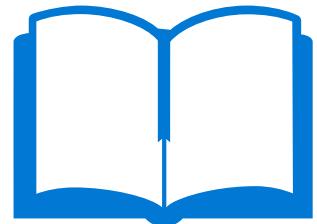
© Copyright Microsoft Corporation. All rights reserved.

289

Microsoft Learn module

Performing common actions with client script

<https://learn.microsoft.com/training/modules/common-actions-client-script-power-platform>



© Copyright Microsoft Corporation. All rights reserved.

290

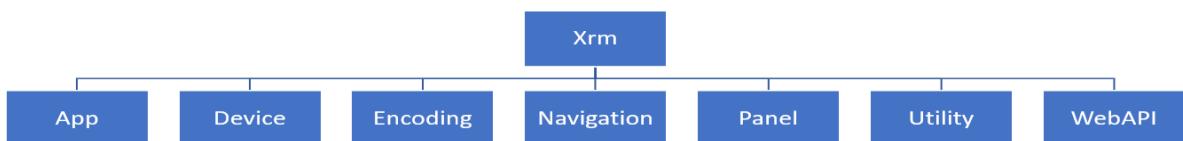
Introduction to client-side scripting

Form events

- Form OnLoad
- Form OnSave
- Column OnChange

Can be used to:

- Get or set column values on the form
- Show and hide user interface elements
- Switch between forms where multiple forms are available for a table
- Open forms and views
- Interact with the business process flow



© Copyright Microsoft Corporation. All rights reserved.

291

Web resources

Upload files to Dataverse

Virtual file system

Use in forms and apps

Use as JavaScript libraries

Types:

- Script (Jscript)
- Webpage (HTML)
- Style Sheet (CSS)
- Data (XML)
- Images
- PNG
- JPG
- GIF
- ICO

© Copyright Microsoft Corporation. All rights reserved.

292

Upload scripts

To use client scripting on a form the script must first be uploaded as a JScript web resource

293

Using client script libraries

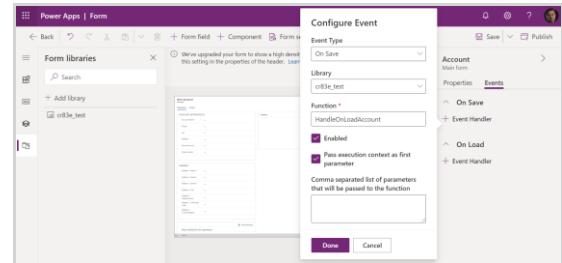
Once configured as a script web resource, client script libraries can be associated with commands and form events

294

Registering via form properties

Registering event handlers using form properties creates a static configuration of event handlers at design time.

- **Form** - This allows you to register **OnLoad** and **OnSave** event handlers.
- **Columns** - This allows you to register an **OnChange** event handler if column data is changed



295

Registering using code

Registering event handlers using code is possible for all handlers except **OnLoad**, which must be registered using configuration

- Create a function that runs in OnLoad of the account form.
- In that function, call addOnChange to register the function to call when the account number column changes.
- Register on the account form properties the OnLoad event handler.

JavaScript

```
function LearnLab_handleAccountOnLoad(executionContext)
{
    var formContext = executionContext.getFormContext();

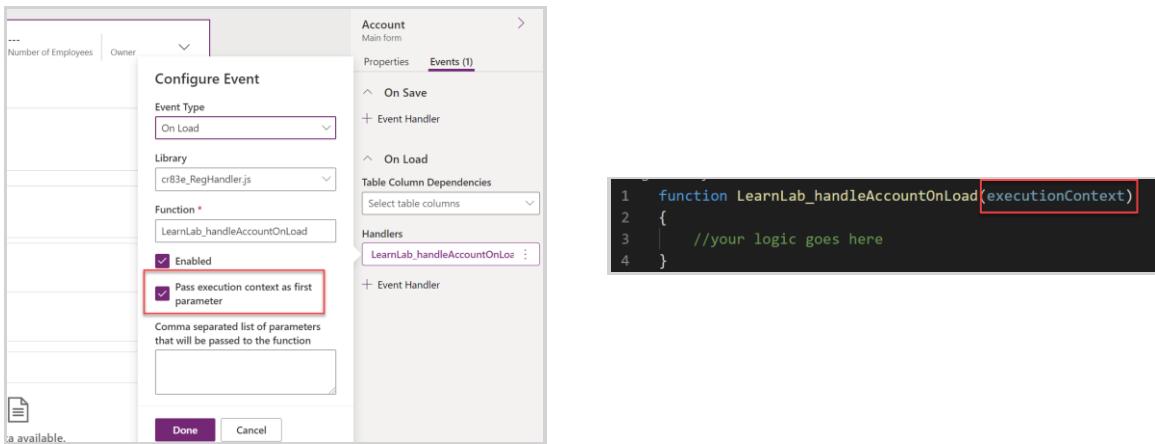
    formContext.getAttribute('accountnumber').addOnChange(LearnLab_
        handleOnChangeAccountNumber)
}

function
LearnLab_handleOnChangeAccountNumber(executionContext)
{
    var formContext = executionContext.getFormContext();
    formContext.ui.setFormNotification('Check other systems', 'INFO',
    'AcctNumber');
}
```

296

Execution context

When you register an event handler, you can have the execution context passed as the first parameter.



297

Client API Object model

executionContext

- Execution context for an event in model-driven apps forms and grids
- Pass executionContext as first parameter

formContext

- Reference to a form
- executionContext.getFormContext method gets the formContext object

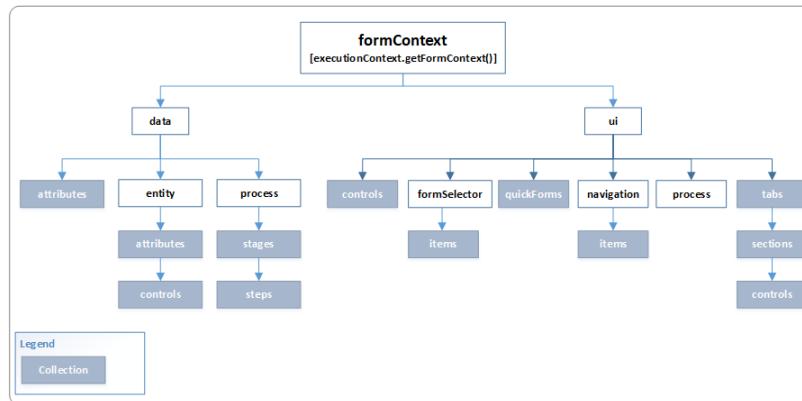
gridContext

- Reference to a grid or a subgrid on a form

Xrm

- Global object for performing operations that do not directly impact the data and UI in the form

Form context



© Copyright Microsoft Corporation. All rights reserved.

299

Form context

The Client API form context (`formContext`) provides a reference to the form or to an item on the form and replaces `Xrm.Page`

```

1  function LearnLab_HandleAccountOnLoad(executionContext)
2  {
3      var formContext = executionContext.getFormContext();
4  }
  
```

JavaScript

```

var firstName =
Xrm.Page.getAttribute("firstname").getValue();

var firstName =
formContext.getAttribute("firstname").getValue();
  
```

300

Form types

JavaScript

```
var formContext =
executionContext.getFormContext();
var formType =
formContext.getFormType();
```

Form types:

- 1 Create
- 2 Update
- 3 Read Only
- 4 Disabled
- 6 Bulk Edit

© Copyright Microsoft Corporation. All rights reserved.

301

Accessing Dataverse column data

Task	Example
Access by name	var nameAttribute = formContext.getAttribute("name"); Assigns the attribute for the Account Name column to the nameAttribute variable. If attribute is not present on the form, getAttribute method returns null value.
Access all attributes	var allAttributes = formContext.getAttribute(); Assigns an array of all the attributes in the formContext.data.entity.attributes collection to the allAttributes variable.

A Dataverse column is represented in the object model as an attribute object.

© Copyright Microsoft Corporation. All rights reserved.

302

Using attributes

Task	Example
Get the value of an attribute	<pre>var nameValue = formContext.getAttribute("name").getValue();</pre> <p>Assigns the value of the Account Name column to the nameValue variable.</p>
Set the value of an attribute	<pre>formContext.getAttribute("name").setValue("new name");</pre> <p>Sets the value of the Account Name column to "new name".</p>
Get the currently selected option object in an OptionSet attribute (OptionSet attribute describes the Dataverse Choice column)	<pre>var addressTypeOption = formContext.getAttribute("address1_addressstypecode").getSelectedOption();</pre> <p>Assigns the selected option in the Address Type column to the addressTypeOption variable.</p>
Determine whether an attribute value has changed in the user interface since the form has been opened	<pre>var isNameChanged = formContext.getAttribute("name").getIsDirty();</pre> <p>Assigns a Boolean value that indicates whether the Account Name column value has changed to the isNameChanged variable.</p>
Change whether data is required in a column in order to save a record	<pre>formContext.getAttribute("creditlimit").setRequiredLevel("required");</pre> <p>Makes the Credit Limit column required. <pre>formContext.getAttribute("creditlimit").setRequiredLevel("none");</pre><p>Makes the Credit Limit column optional.</p></p>
Determine whether the data in an attribute will be submitted when the record is saved	<pre>var nameSubmitMode = formContext.getAttribute("name").getSubmitMode();</pre> <p>The nameSubmitMode variable value will be either always, never, or dirty text to represent the submitMode for the Account Name column.</p>
Control whether data in an attribute will be saved when the record is saved	<pre>formContext.getAttribute("name").setSubmitMode("always");</pre> <p>The example will force the Account Name column value to always be saved even when it has not changed.</p>
When column level security has been applied to an attribute, determine whether a user has privileges to perform create, read, or update operations on the attribute.	<pre>var canUpdateNameAttribute = formContext.getAttribute("name").getUserPrivilege().canUpdate();</pre> <p>Assigns a Boolean value that represents the user's privilege to update the Account Name column to the canUpdateNameAttribute variable.</p>

303

Accessing form controls

Task	Example
Access all the controls for a specific attribute	<pre>var nameControls = formContext.getAttribute("name").controls.get();</pre> <p>Assigns an array of all the controls for the name attribute to the nameControls variable.</p>
Access a control by name	<pre>var nameControl = formContext.getControl("name");</pre> <p>The first control added to a form for a column will have the same name as the column. Each additional control name will have an index number appended to the name. For example, three controls for the name column will have the names: name, name1, and name2 respectively.</p>
Access all controls	<pre>var allControls = formContext.getControl();</pre> <p>Assigns an array of all the controls in the formContext.ui.controls collection to the allControls variable.</p>

304

Use form controls

Control objects like attribute objects have a common set of methods regardless of the type.

Task	Example
Determine if a control is visible	<code>var isNameVisible = formContext.getControl("name").getVisible();</code> Assigns a Boolean value to the isNameVisible variable that represents whether the Account Name column is visible.
Hide or show a control	<code>formContext.getControl("name").setVisible(false);</code> Hides the Account Name column.
Get a reference to the attribute for the control	<code>var nameAttribute = formContext.getControl("name").getAttribute();</code> Assigns the attribute for the control for the Account Name column to the nameAttribute variable.
Disable or enable all controls for an attribute	<code>formContext.getAttribute("name").controls.forEach(function (control, index) { control.setDisabled(true); });</code>
Change the label for a control	<code>formContext.getControl("name").setLabel("Company Name");</code> Sets the label for the Account Name column to the text Company Name.
Get the parent of a control	<code>var parentSection = formContext.getControl("name").getParent();</code> Assigns the parent control of the Account Name column to parentSection variable.
Set focus on a control	<code>formContext.getControl("name").setFocus();</code> Set current input focus to the Account Name column.

305

Use tabs and sections

Each form has collection of tabs.

Task	Example
Show or hide a tab	<code>formContext.ui.tabs.get("general").setVisible(false);</code> Hides general tab.
Change the label for a tab	<code>formContext.ui.tabs.get("general").setLabel("Major");</code> Sets the label of the general tab to the text Major.
Show or hide a section	<code>formContext.getControl("industrycode").getParent().setVisible(false);</code> Hides section containing the Industry Code column.

306

Use entity data

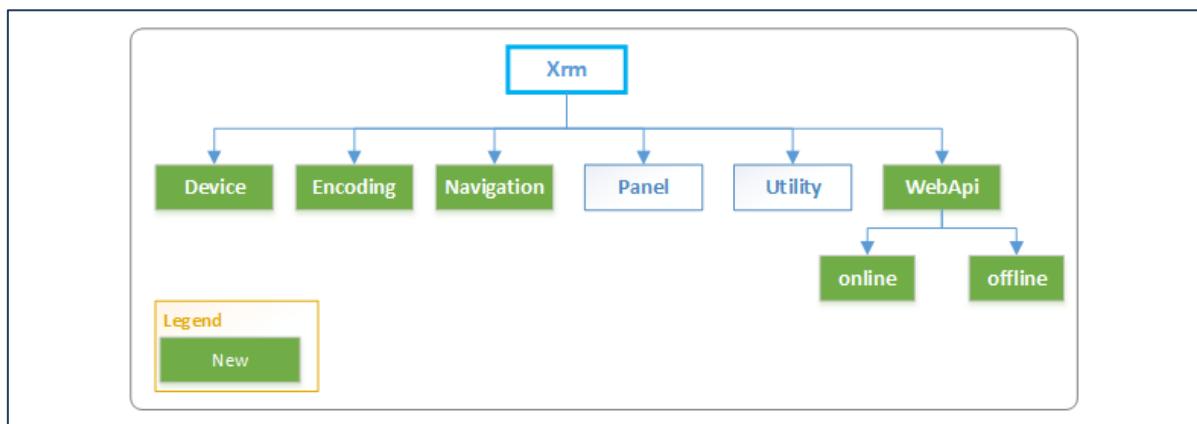
The following table contains methods you can use to get information about the current record.

Task	Example
Get the ID of the current record	<code>var recordId = formContext.data.entity.getId();</code> Assigns a current record unique identifier to the recordId variable. If the form is opened to create a new record null value is returned.
Save the current record	<code>formContext.data.entity.save();</code> Use saveandclose or saveandnew to perform the equivalent actions using the Save & Close or Save & New
Determine whether any data in the current row is changed.	<code>var isDirty = formContext.data.entity.getIsDirty();</code> Assigns a Boolean value that indicates whether any column value on the form has changed to the isDirty variable.

307

Introduction to conducting global operations with the client API Xrm object

The Client API provides a globally accessible object (Xrm), which is available for use in your code to perform a variety of activities.



© Copyright Microsoft Corporation. All rights reserved.

308

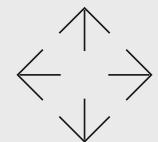
Xrm.WebApi object

The Xrm.WebApi object provides properties and methods to use the Web API for traditional CRUD operations within a client script.

Method	Description
<code>createRecord</code>	Creates a table row.
<code>deleteRecord</code>	Deletes a table row.
<code>retrieveRecord</code>	Retrieves a table row.
<code>retrieveMultipleRecords</code>	Retrieves a collection of table rows.
<code>updateRecord</code>	Updates a table row.
<code>isAvailableOffline</code>	Returns a Boolean value that indicates whether a table is present in a user's profile and is currently available for use in offline mode.
<code>execute</code>	Run a single action, function, or CRUD operation.
<code>executeMultiple</code>	Run a collection of action, function, or CRUD operations.

309

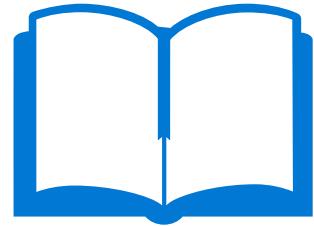
Module 2: Best practices with client script



Microsoft Learn module

Automate business process flows with client script

<https://learn.microsoft.com/training/modules/automate-business-process-flow-client-script-power-platform>



© Copyright Microsoft Corporation. All rights reserved.

311

Client scripting vs. Business rules

JavaScript	Business rules
<ul style="list-style-type: none">• Model-driven app form only• Operates on columns, sections, and tabs• Use Web API to access any table/column• Use with command buttons• Supports complex logic• Supports OnSave• Notifications	<ul style="list-style-type: none">• Model-driven app form and Dataverse• Operates on columns only• Limited to columns on table/form• Show error next to field• Events OnLoad and OnChange only• Recommendations

© Copyright Microsoft Corporation. All rights reserved.

312

Business process flows and JavaScript and Business rules

Actions on a column with a Business rule or with JavaScript performs the same action on the corresponding step in a Business Process flow

- SetVisibility
- SetValue

© Copyright Microsoft Corporation. All rights reserved.

313

Calculated columns vs. client script

Client script are actioned immediately in the user interface.

Calculated columns are calculated on retrieve, so client changes won't show until data is refreshed.

© Copyright Microsoft Corporation. All rights reserved.

314

Coding Standards and Best Practices

Define unique Script function names

Unique Function Prefix

JavaScript

```
function MyUniqueName_performMyAction()
{
    // Code to perform your action.
}
```

© Copyright Microsoft Corporation. All rights reserved.

315

Coding Standards and Best Practices

Define unique Script function names

Namespaced Library Names

JavaScript

```
//If the MyUniqueName namespace object isn't defined, create it.
if (typeof (MyUniqueName) == "undefined")
{ MyUniqueName = {}; }
// Create Namespace container for functions in this library;
MyUniqueName.MyFunctions = { performMyAction: function(){
    // Code to perform your action. //Call another function in your library
    this.anotherAction(); },
anotherAction: function(){
    // Code in another function } };
```

When you use your function, you can specify the full name, as shown in the following example.
`MyUniqueName.MyFunctions.performMyAction();`

© Copyright Microsoft Corporation. All rights reserved.

316

Debugging client script

Nearly every browser has some sort of debugging extension that allows for debugging of client scripts.

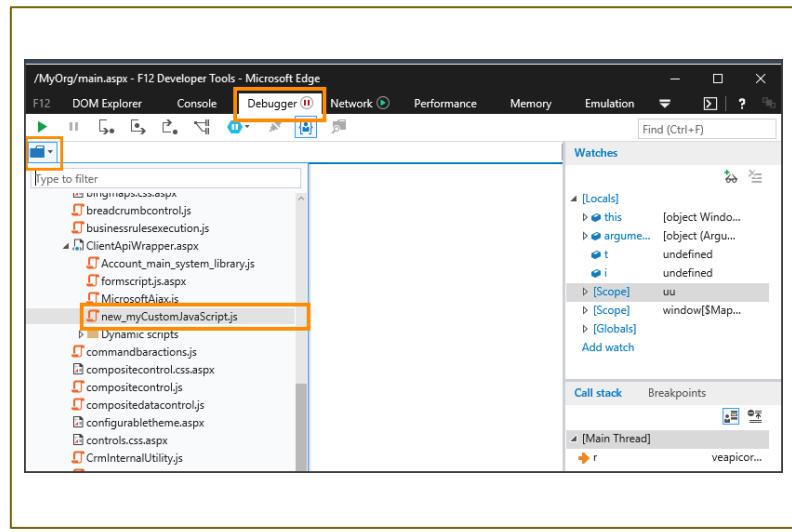
- Microsoft Edge (through F12 Developer Tools).
- Google Chrome (through F12 Developer Tools)
- Mozilla Firefox (using Firebug)
- Apple Safari (using Web Inspector)

© Copyright Microsoft Corporation. All rights reserved.

317

Viewing script resources

When your model-driven app form page is loaded, all client script libraries are loaded alongside the webpage as individual script resources.



© Copyright Microsoft Corporation. All rights reserved.

318

Write messages to the console

Using the [window.alert method](#) when debugging JavaScript is still a common way to troubleshoot code in the application – instead use [console.log](#)

JavaScript

```
function writeToConsole(message)
{
  if (typeof console != 'undefined') {
    console.log(message);
  }
}
```

319

Use Fiddler Auto-Responder to replace web resource content

Constantly editing web resources when they are under development can prove to be difficult and time-intensive when you have to republish the files within a solution upon every edit.

320

Learning Path 7 practice labs

Extending the model-driven app user experience

Lab 8: Client scripting



© Copyright Microsoft Corporation. All rights reserved.

321

Microsoft Dynamics 365

© Copyright Microsoft Corporation. All rights reserved.

322



Learning Path 8: Create code components with the Power Apps Component Framework

© Copyright Microsoft Corporation. All rights reserved.



323

Module 1: Get started with Power Apps component framework



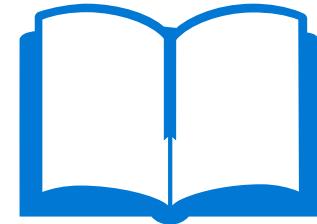
© Copyright Microsoft Corporation. All rights reserved.

324

Microsoft Learn module

Get started with Power Apps component framework

<https://learn.microsoft.com/training/modules/get-started-component-framework>



© Copyright Microsoft Corporation. All rights reserved.

325

Introduction to Power Apps component framework

Microsoft Power Apps component framework helps you create reusable components that can be used within your Power Apps applications.

The screenshot shows a Microsoft Dynamics 365 Sales Opportunity record for "4G Enabled Tablets". The record details include:

- Topic:** 4G Enabled Tablets
- Contact:** Vincent Laurent
- Account:** Southbridge Video
- Purchase Timeframe:** This Quarter
- Currency:** US Dollar
- Budget Amount:** \$4,932,000.00
- Purchase Process:** Individual
- Description:** ---

The opportunity has a status of "In Progress" and an estimated close date of 3/6. Estimated revenue is \$3,257,500.00.

The timeline shows the following stages:

- Qualify
- Develop (Current Stage)
- Propose (5 d)

Recent activity in the timeline includes:

- A task completed by Spencer Low (Sample Data) on Saturday 6:39 PM.
- A phone call from Alpine Ski House on Saturday 6:38 PM.

The Relationship Assistant pane shows Vincent Laurent as a stakeholder. The Stakeholders pane also lists Vincent Laurent.

© Copyright Microsoft Corporation. All rights reserved.

326

Introduction to Power Apps component framework

However, if you reconfigured your app to use custom Power Apps components, your app might look something like the following image.



© Copyright Microsoft Corporation. All rights reserved.

327

What are PCF Code components



Reusable rich user interface controls



Created by developers



Developed using HTML, CSS, and TypeScript



Packaged as solutions



Can be used in model-driven apps, canvas apps, and Power Pages sites

© Copyright Microsoft Corporation. All rights reserved.

328

Power Apps code component framework advantages

A few of the advantages that you are afforded as a result are:

-  Access to a rich set of framework APIs that expose capabilities like component lifecycle management, contextual data, and metadata
-  Support of client frameworks such as React and AngularJS
-  Seamless server access through Web API, utility and data formatting methods, device features like camera, location, and microphone, in addition to easy-to-invoke UX elements like dialogs, lookups, and full-page rendering
-  Optimization for performance
-  Reusability
-  Use of responsive web design principles to provide an optimal viewing and interaction experience for any screen size, device, or orientation
-  Ability to bundle all files into a single solution file

© Copyright Microsoft Corporation. All rights reserved.

329

Types of components that you can add

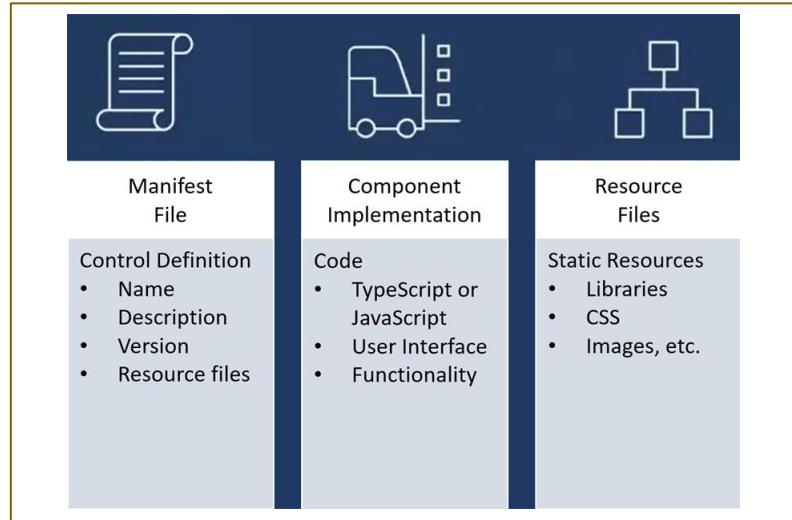
-  Field: A custom control for a column on a form
-  Dataset: A custom grid (or subgrid) to display data in a tabular format
-  A component that displays content from external services

© Copyright Microsoft Corporation. All rights reserved.

330

Component composition

Code components are implemented using HTML, CSS, and TypeScript



© Copyright Microsoft Corporation. All rights reserved.

331

Power Apps component life cycle

When developing a component, you are expected to implement the life cycle methods that are shown in the following table.

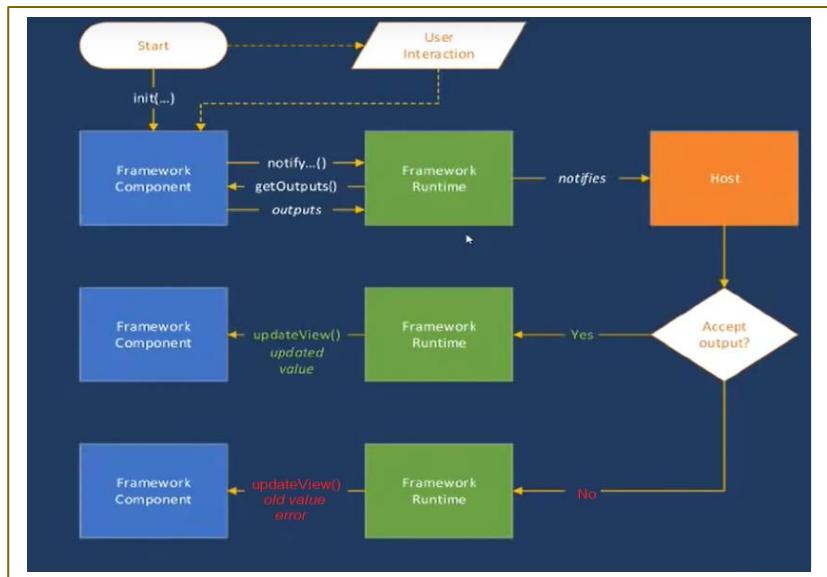
Method	Description
init	This method is used to initialize the component instance. Components can kick off remote server calls and other initialization actions in this method. Dataset values cannot be initialized with this method; you will need to use the updateView method for this purpose.
updateView	This method will be called when any value in the component's property bag has changed.
getOutputs	Called by the framework prior to the receipt of new data. Use this method when dynamically managing bound properties in a control. (Optional)
destroy	Invoked when the component is to be removed from the DOM tree. Used for cleanup and to release any memory that the component is using.

© Copyright Microsoft Corporation. All rights reserved.

332

Power Apps component life cycle

These methods are invoked through a Framework Runtime process in a standardized life cycle, as shown in the following illustration.



© Copyright Microsoft Corporation. All rights reserved.

333

Power Apps component tooling

Power Apps CLI

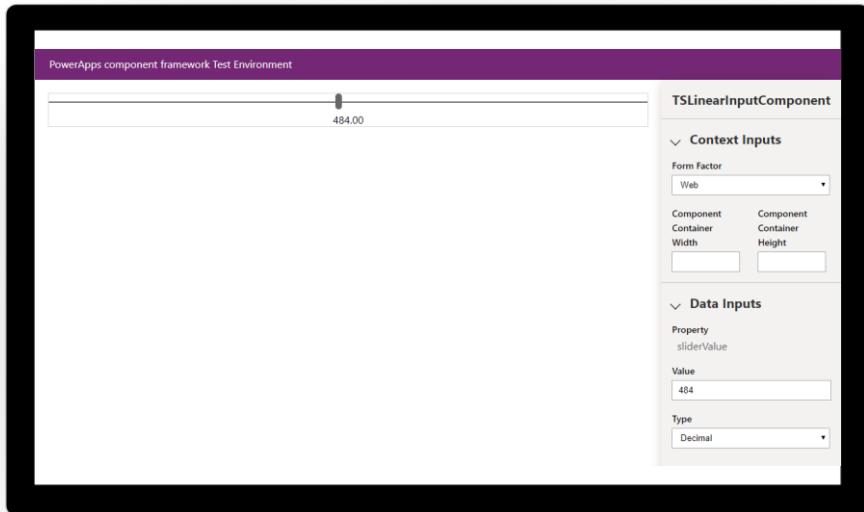
Power Apps
component CLI
commands

Choose an IDE

© Copyright Microsoft Corporation. All rights reserved.

334

Test and debug code components



© Copyright Microsoft Corporation. All rights reserved.

335

Learning Path 8 practice labs

Power Apps Component Framework code components

Lab 9: Power Apps Component Framework (Optional)



© Copyright Microsoft Corporation. All rights reserved.

336



© Copyright Microsoft Corporation. All rights reserved.

337



Learning Path 9: Extending Microsoft DataVerse

© Copyright Microsoft Corporation. All rights reserved.



338

Module 1: Introduction to Dataverse for developers



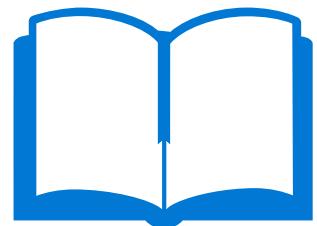
© Copyright Microsoft Corporation. All rights reserved.

339

Microsoft Learn module

Introduction to Dataverse for developers

<https://learn.microsoft.com/training/modules/intro-cds-developers-power-platform>

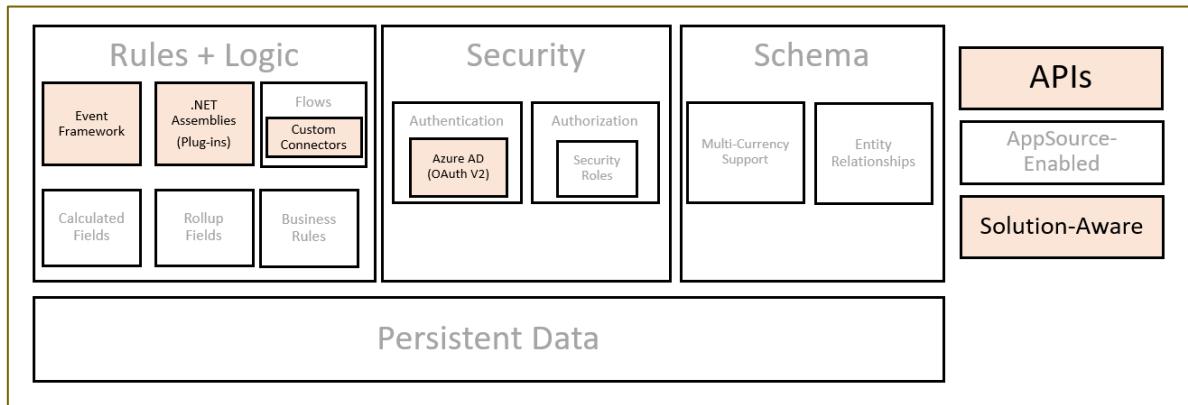


© Copyright Microsoft Corporation. All rights reserved.

340

Dataverse extensibility model

To better understand how to extend Dataverse, it is valuable to learn about its underlying architecture.

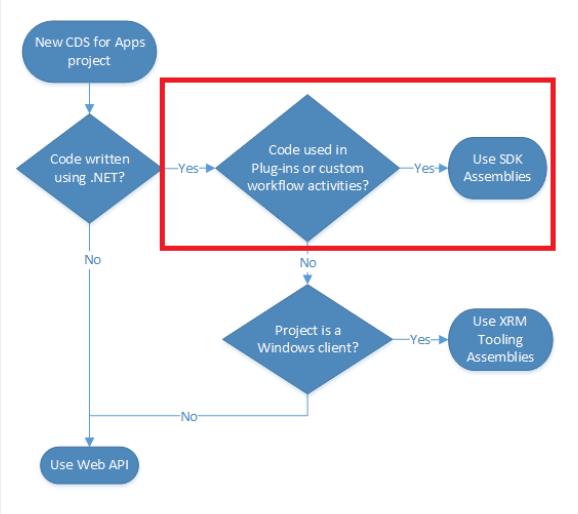


© Copyright Microsoft Corporation. All rights reserved.

341

.NET Assemblies (plug-ins)

Plug-ins are managed code classes that are registered to run when specific events occur.

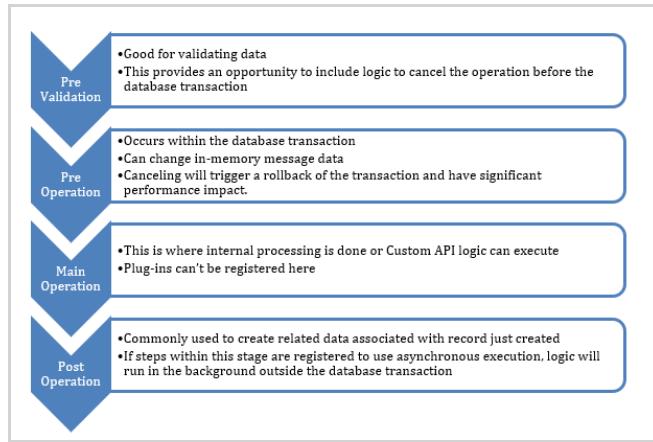


© Copyright Microsoft Corporation. All rights reserved.

342

Event pipeline

When you perform an action like create a record in an app or create a record using the API, a Create message is processed by Dataverse.



343

Execution modes (Synchronous versus Asynchronous)

Plug-ins can be configured to run synchronously or asynchronously, depending on what type of pipeline operation we are handling.

Plug-ins registered in synchronous mode will run as soon as their execution pipeline stage is reached, and the entire operation will not proceed until the logic has completed execution.

Plug-ins registered in asynchronous mode will be dispatched as a system job to the Asynchronous service, which executes their logic after the given operation completes.

Module 2: Create plug-ins



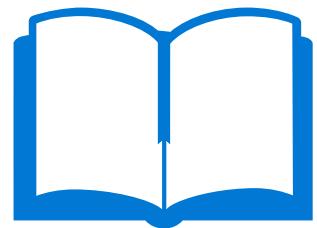
© Copyright Microsoft Corporation. All rights reserved.

345

Microsoft Learn module

Extend plug-ins

<https://learn.microsoft.com/training/modules/plug-ins-extend-power-platform>



© Copyright Microsoft Corporation. All rights reserved.

346

Introduction to Plug-ins

A plug-in is *imperative* logic that should be used only when a *declarative* process, such as a business rule, flow, or workflow, does not meet your requirement.

- Canceling the event and displaying an error to the user.
- Making changes to the data in the operation.
- Initiating other actions by using the Organization Service to add automation.

© Copyright Microsoft Corporation. All rights reserved.

347

Alternatives to plug-ins

Plug-ins should be considered a last resort in many cases.

- Dataverse workflows
- Power Automate cloud flows
- Calculated and rollup columns
- Custom actions

© Copyright Microsoft Corporation. All rights reserved.

348

Plug-in considerations

Plug-ins perform better when you consider their performance, capabilities, and ability to run synchronously.

A well-written plug-in will always be the most efficient way to apply business logic to Dataverse.

Plug-ins provide several capabilities that are not available with declarative business logic, such as efficiently working with external services in code.

If synchronous logic is required for your application, plug-ins might be a required option for you.

© Copyright Microsoft Corporation. All rights reserved.

349

Workflows/flows vs. plug-ins/client script

Circumstances might occur when existing limitations require you to develop plug-ins to accomplish certain activities.

	Workflow	Plug-in	Client Script
Synchronous	Either	Either	Synchronous
Access External Data	No	Yes	Yes (with limitations)
Maintenance	Business Users	Developers	Developers
Can Run As	User	Any licensed user or current user	User
Can Run On Demand	Yes	No	No
Can Nest Child Processes	Yes	Yes	No
Execution Stage	Before/After	Before/After	Before/After
Triggers	Create, Column Change, Status Change, Assign to Owner, On Demand	Create, Column Change, Status Change, Assign to Owner, Delete, along with many other specialized triggers	Column Change or Form Load

350

Building plugins

Plugins are .NET classes that implement a **IPlugin** interface provided by the Dataverse SDK assemblies.

- **IPluginExecutionContext** - this gives you access to the message being processed and information about the requestor.
- **ITracingService** - this gives you access to write to the trace log for diagnostic purposes.
- **IOrganizationServiceFactory** - this gives you access to retrieve an **OrganizationService** for use in accessing data from the plug-in.

351

Plug-in execution context

Event Execution pipeline passes information about the current operation in the **IPluginExecutionContext** interface to the plug-in:

- PrimaryEntityId
- PrimaryEntityName
- MessageName
- Stage
- Depth
- InputParameters
- OutputParameters
- PreEntityImages
- PostEntityImages

```
IPPluginExecutionContext context =
    (IPPluginExecutionContext)serviceProvider.GetService(
        typeof(IPPluginExecutionContext));
var entity =
    (Entity)context.InputParameters["Target"];
```

352

Develop a plug-in

Inherit from IPlugin interface

Execute method

Extract execution content

Access Organization service

Impersonation

- context.UserId
- context.InitiatingUserId

```
public class MyPlugin : IPlugin
{
    public void Execute(IServiceProvider serviceProvider)
    {
        IPluginExecutionContext context =
            (IPluginExecutionContext)serviceProvider.GetService(typeof(IPluginExecutionContext));

        IOrganizationServiceFactory serviceFactory =
            (IOrganizationServiceFactory)serviceProvider.GetService(typeof(IOrganizationServiceFactory));
        IOrganizationService orgService =
            serviceFactory.CreateOrganizationService(context.UserId);
    }
}
```

© Copyright Microsoft Corporation. All rights reserved.

353

Error handling

Organization service errors generate ServiceFaults

To show a message to the user create an InvalidPluginExecutionException

```
try { }
catch (FaultException<OrganizationServiceFault> ex)
{
    throw new InvalidPluginExecutionException("The
        following error occurred in MyPlugin.", ex);
}
catch (Exception ex)
{
    tracingService.Trace("MyPlugin: error: {0}",
        ex.ToString());
    throw;
}
```

© Copyright Microsoft Corporation. All rights reserved.

354

Tracing service

Tracing Service

Enable trace logging

- Off
- Exceptions
- All

```
ITracingService tracingService =  
(ITracingService)serviceProvider.GetService(typeof  
(ITracingService));
```

```
tracingService.Trace("Write {0} {1}", "your",  
"message");
```

© Copyright Microsoft Corporation. All rights reserved.

355

Develop a plug-in

PluginBase abstract class

Power Platform Tools for Visual Studio

- Power Platform Plug-in Library template

Power Platform CLI

Run CLI command

```
pac plugin init
```

© Copyright Microsoft Corporation. All rights reserved.

356

InputParameters

Input parameters are contained in the execution context's InputParameters collection and are of type *Entity*.

- Register your plug-in to run on the PreValidation stage on Create and/or Update of the table that you want to validate.
- Validate data in your plug-in by reading the values from the InputParameters collection. On Create, you'll want to retrieve the Target collection.
- Throw an InvalidPluginExecutionException if the provided data is not valid.

© Copyright Microsoft Corporation. All rights reserved.

357

OutputParameters

Output parameters are contained in the execution context's OutputParameters collection and are of the type *Table*.

- Register your plug-in to run on the PostOperation stage on Create and/or Update of the table that you want to update.
- Update the data in your plug-in by editing the values from the OutputParameters collection. Note that on Create, you'll want to retrieve the Target collection.

© Copyright Microsoft Corporation. All rights reserved.

358

Pre and Post Entity Images

- Allow plug-in code to reference columns not in the operation
- Before (Pre) and after (Post) values
- Prevent the need to retrieve a row as first step in plug-in

© Copyright Microsoft Corporation. All rights reserved.

359

Debug and troubleshoot a plug-in

- ITracing service
- Raise InvalidPluginExecution Exception to display message to user
- Plug-in Trace logs
- System jobs for asynchronous plug-ins
- Plug-in profiler
- Analyze plug-in performance with Analytics in the Power Platform admin center
- Timeout: 2 minute limit
- Error handling

© Copyright Microsoft Corporation. All rights reserved.

360

Optimize plug-in performance

- Filter the columns to trigger plug-in so only run when required
- Use Pre and Post image to remove need to retrieve row as first step
- Do not register plug-in on any table
- Minimize operations as 2 minute limit
- Check the depth of the transaction and abort if necessary
- Handle all errors

© Copyright Microsoft Corporation. All rights reserved.

361

Configure a Custom API message

Create own messages in Dataverse APIs

Tables

- Custom API
- Custom API Request Parameter
- Custom API Response Property

Global or Bound to a table

Create with

- Power Apps maker portal
- Plugin Registration Tool
- Code

© Copyright Microsoft Corporation. All rights reserved.

362

Develop a plug-in that implements a Custom API

- Add plug-in to perform logic when custom API is called
- Write code just like you would any other plug-in
- Register the assembly
- Select the assembly Custom API Plugin Type in the Custom API record
- Do not need to register a step

© Copyright Microsoft Corporation. All rights reserved.

363

Learning Path 9 practice labs

Extending Microsoft Dataverse

Lab 10: Dataverse Plug-ins



© Copyright Microsoft Corporation. All rights reserved.

364



© Copyright Microsoft Corporation. All rights reserved.

365



Learning Path 10: Integrate Dataverse and Azure

© Copyright Microsoft Corporation. All rights reserved.



366

Module 1: Integrate with Azure



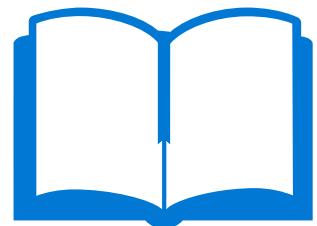
© Copyright Microsoft Corporation. All rights reserved.

367

Microsoft Learn module

Integrate Dataverse Azure solutions

<https://learn.microsoft.com/training/modules/integrate-common-data-service-azure-solutions>



© Copyright Microsoft Corporation. All rights reserved.

368

Native Power Platform Azure integrations

Azure Active Directory

- Authentication
- Register apps

Azure Service Bus

Azure Event Hub

Logic apps

- Microsoft Dataverse connector

Azure data lake and Azure Synapse

Analytics

- Azure Synapse link

Azure Bot Service

- Skills to link to PVA bots

Virtual tables

- Azure SQL
- Cosmos DB

Connectors to Azure Services

- Azure SQL
- Azure Storage
- Cognitive Services

© Copyright Microsoft Corporation. All rights reserved.

369

Comparing Azure's integration solutions

Logic Apps - Azure Logic Apps provides a robust visual interface in which you can orchestrate complex integrations across your various environments.

Azure Service Bus - Azure Service Bus is Microsoft's cloud messaging as a service (MaaS) platform.

Azure Functions – Serverless code that can be triggered from Dataverse and other Azure services

Event Grid - Microsoft Azure Event Grid is an event-driven, publish-subscribe framework that allows you to handle various events.

Event Hubs - Azure Event Hubs is Microsoft's version of Apache Kafka and provides a real-time data ingestion service that supports millions of events per second.

© Copyright Microsoft Corporation. All rights reserved.

370

Dataverse outbound integration with Azure

Azure Service Bus	Azure Event Hub	Webhook
No code and code integration	No code and code integration	Code and no code integration
Post messages to Service Bus Queues, Topics, Relays	Post messages and events to Event Hub	Simple HTTP trigger
Post Operation only	Post Operation only	

© Copyright Microsoft Corporation. All rights reserved.

371

Choosing the right Azure integration solution

IF YOU WANT TO...	USE THIS
Perform workflow and orchestrate within and across systems at scale and with monitoring and error handling	Logic Apps
Connect on-premises and cloud systems or decouple transactional systems	Service Bus
Process streaming data	Event Hub
Publish and subscribe to events	Event Grid
Send event data to external HTTP endpoints	Webhook
Allow external systems to read and write Dataverse data	Custom API

372

Integration with Azure Service Bus Queue

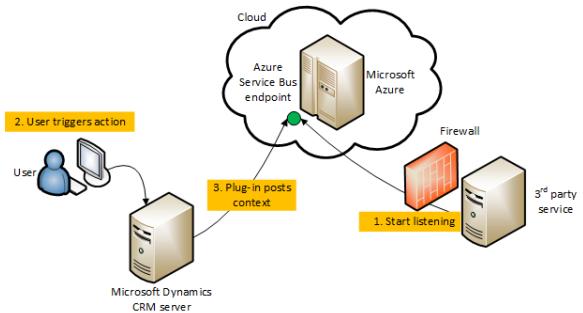
Generate Shared Access Signature (SAS) token

Send execution context as

- Binary
- JSON
- XML

Queue Listener

- Logic app
- Azure Function
- Custom code



© Copyright Microsoft Corporation. All rights reserved.

373

Publish event using the IServiceProviderNotificationService

- Create a plug-in
- Requires the ID of the service endpoint
- Use IServiceProviderNotificationService interface
- Execute using the serviceendpoint table
- Pass your custom message as second parameter

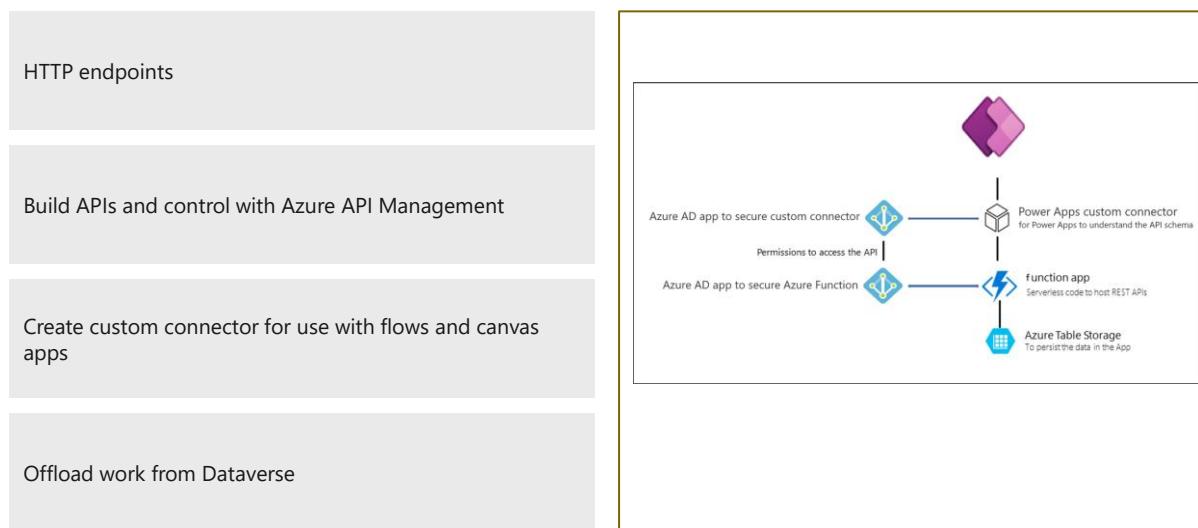
```
public void Execute(IServiceProvider serviceProvider)
{
    IServiceProviderNotificationService
    cloudService =
    (IServiceProviderNotificationService)serviceProvider.GetService(typeof(IServiceProviderNotificationService));

    var response = cloudService.Execute(new
    EntityReference("serviceendpoint",
    serviceEndpointId), context);
}
```

© Copyright Microsoft Corporation. All rights reserved.

374

Azure functions



© Copyright Microsoft Corporation. All rights reserved.

375

Discover Azure Functions

- Azure Functions are a great solution for processing data, integrating systems, working with the internet-of-things (IoT), and building simple APIs and microservices.
- Consider Functions for tasks like image or order processing, file maintenance, or for any tasks that you want to run on a schedule.
- Azure Functions supports *triggers*, which are ways to start execution of your code, and *bindings*, which are ways to simplify coding for input and output data.
- Azure Function with HTTP trigger can be called using a Webhook



© Copyright Microsoft Corporation. All rights reserved.

376

Compare Azure Functions hosting plans

The following is a summary of the benefits of the three main hosting plans for Functions:

Consumption plan	Premium plan	Dedicated plan
<p>Consumption plan</p> <ul style="list-style-type: none"> The default hosting plan. Scales automatically and you only pay when your functions are running. Instances of the Functions host are dynamically added and removed based on the number of incoming events. 	<p>Premium plan</p> <ul style="list-style-type: none"> Automatically scales based on demand using pre-warmed workers. Runs on more powerful instances and connects to virtual networks. 	<p>Dedicated plan</p> <ul style="list-style-type: none"> Run your functions within an App Service plan at regular App Service plan rates. Best for long-running scenarios where Durable Functions can't be used.

© Copyright Microsoft Corporation. All rights reserved.

377

Compare Azure Functions hosting plans

Function app timeout duration

- The `functionTimeout` property in the `host.json` project file specifies the timeout duration.
- Consumption** plan has a default timeout of 5 minutes, and a maximum timeout of 10 minutes.
- Premium** and **Dedicated** plans have a default timeout of 302 minutes and no maximum duration.

Storage account requirements

- On any plan, a function app requires a general Azure Storage account.
- Azure Functions relies on Azure Storage for operations such as managing triggers and logging function executions.

© Copyright Microsoft Corporation. All rights reserved.

378

Azure Functions development

A function contains two important pieces:

- **Your code**, which can be written in a variety of languages
- **Some config**, the *function.json* file.

The *function.json* file defines the function's trigger, bindings, and other configuration settings. Every function has one and only one trigger. The runtime uses this config file to determine the events to monitor and how to pass data into and return data from a function execution.

```
{
  "disabled":false,
  "bindings":[
    // ... bindings here
    {
      "type": "bindingType",
      "direction": "in",
      "name": "myParamName",
      // ... more depending on binding
    }
  ]
}
```

© Copyright Microsoft Corporation. All rights reserved.

379

Azure Functions triggers and bindings

- Triggers are what cause a function to run. A trigger defines how a function is invoked and a function must have exactly one trigger.
- Binding to a function is a way of declaratively connecting another resource to the function; bindings may be connected as input bindings, output bindings, or both.
- You can mix and match different bindings to suit your needs.
- Triggers and bindings let you avoid hardcoding access to other services.

© Copyright Microsoft Corporation. All rights reserved.

380

Azure Functions vs. Dataverse Plug-ins

Azure Functions

- Consumption plan 5-minute default and can increased to 10-minute
- Durable functions allow chaining of functions
- Premium and Dedicated plans have longer
- Need to Authenticate

Dataverse Plug-in

- Event driven
- Cannot schedule
- 2-minute limit
- Authenticated automatically

© Copyright Microsoft Corporation. All rights reserved.

381

Publish Dataverse events using webhooks

Another method for publishing events from Dataverse to an external service is to register a webhook

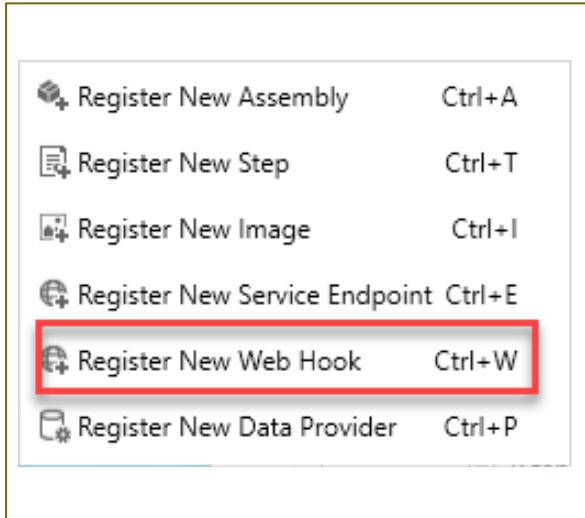
- Name
- Endpoint URL
- Authentication

© Copyright Microsoft Corporation. All rights reserved.

382

Register a Webhook endpoint

Webhook endpoint registration is performed similarly to Service Endpoint registration, by using the Plug-in Registration Tool.



© Copyright Microsoft Corporation. All rights reserved.

383

Webhook authentication options

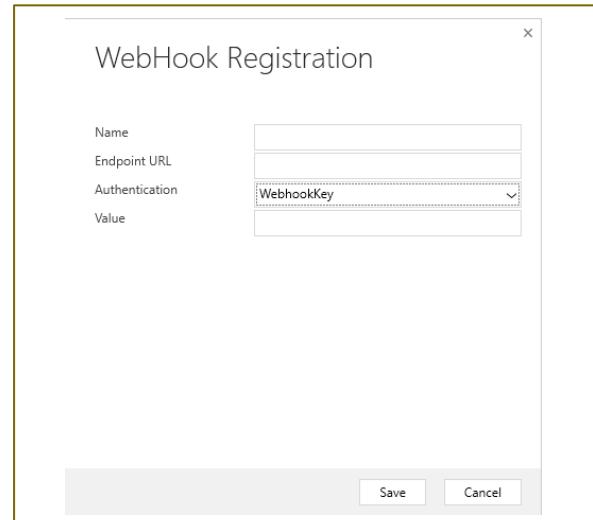
The following table describes the three authentication options that you can use to consume a webhook message from a given endpoint.

Type	Description
HttpHeader	Includes one or more key value pairs in the header of the HTTP request. Example: Key1: Value1, Key2: Value2
WebhookKey	Includes a query string by using code as the key and a value that is required by the endpoint. When registering the webhook by using the Plug-in Registration Tool, only enter the value. Example: ?code=00000000-0000-0000-0000-000000000001
HttpQueryString	Includes one or more key value pairs as query string parameters. Example: ?Key1=Value1&Key2=Value2

384

Register with WebhookKey Authentication

If **WebhookKey** is specified as the **Authentication** method, a query string is passed to the URL with the given key in the format ?code=[web hook key].



© Copyright Microsoft Corporation. All rights reserved.

385

Webhooks vs. Azure Service Bus

Service Bus

- Asynchronous
- Register with Plugin Registration Tool
- Invoke with no code or with code
- Requests can be queued

Webhook

- Synchronous or Asynchronous
- Register with Plugin Registration Tool
- Invoke with no code or with code
- Requests dependent on HTTP endpoint being available

© Copyright Microsoft Corporation. All rights reserved.

386

Learning Path 10 practice labs

Extending Microsoft Dataverse

Lab 11: Azure Functions
(Optional)



Lab 12: Publishing events
externally



© Copyright Microsoft Corporation. All rights reserved.

387

Module 2: Integrate with Dataverse



© Copyright Microsoft Corporation. All rights reserved.

388

Integrate with Dataverse

There are several features of Dataverse that external systems can use to integrate with Dataverse:

- Change tracking
- Alternate keys
- Upsert method

© Copyright Microsoft Corporation. All rights reserved.

389

Change Tracking

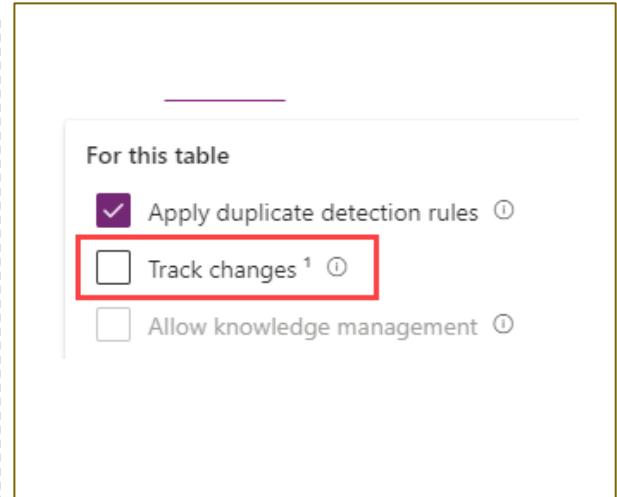
- Pulls changed records from Dataverse
- Enable on table metadata
- Note: Change Tracking is not associated with Auditing
- Retrieve list of changes since last request using a token
- Set Token to null for initial extract
- Store token and use to call to next time
- Multiple Tokens allowed
- Tokens expire after 30 days by default

© Copyright Microsoft Corporation. All rights reserved.

390

Enable change tracking for a table

You can enable change tracking on a table by setting the **Track changes** flag on the table properties in the Power Apps maker portal



© Copyright Microsoft Corporation. All rights reserved.

391

Read table change records by using platform APIs

Organization Service

- RetrieveEntityChanges message
- Loop through EntityChanges.Changes
- Change types
 - NewOrUpdatedItem
 - RemoveOrDeleted

Web API

- Specify odata.track-changes preference in the request
- Add deltatoken to URL parameters
- Returns deltaLink

© Copyright Microsoft Corporation. All rights reserved.

392

Retrieve changes in tables by using the Web API

OData

```
GET [Organization
URI]/org1/api/data/v9.
1/accounts?$select=name,
accountnumber,telephone1,
fax HTTP/1.1
Prefer: odata.track-
changes
Cache-Control: no-
cache
OData-Version: 4.0
Content-Type:
application/json
```

OData

```
{
    "@odata.context": "[Organization
URI]/data/v9.0/$metadata#accounts(name,telephone1,fax)/*delta",
    "@odata.deltaLink": "[Organization
URI]/api/data/v9.1/accounts?$select=name,telephone1,fax&$deltatoken=919058%2108%2f22
%2f2017%2008%3a21%3a20",
    "value":
        [
            {
                "@odata.etag": "W/\"915244\",
                "name": "Monte Orton",
                "telephone1": "555000",
                "fax": "10101",
                "accountid": "60c4e274-0d87-e711-80e5-00155db19e6d"
            },
            {
                "@odata.context": "[Organization
URI]/api/data/v9.0/$metadata#accounts/$deletedEntity",
                "id": "2e451703-c686-e711-80e5-00155db19e6d",
                "reason": "deleted"
            }
        ]
}
```

393

Retrieve changes in tables by using the Web API

OData

```
GET [Organization
URI]/api/data/v9.1/accounts?$select=name,accountnumber,telephone1,fax&$deltatoken=919042%2108%2f22%2f2017%2008%3a10%3a44
```

```
{
    "@odata.context": "[Organization URI]/data/v9.1/$metadata#accounts(name,telephone1,fax)/*delta",
    "@odata.deltaLink": "[Organization URI]/api/data/v9.1/accounts?$select=name,telephone1,fax&$deltatoken=919058%2108%2f22%2f2017%2008%3a21%3a20",
    "value":
        [
            {
                "@odata.etag": "W/\"915244\",
                "name": "Monte Orton",
                "telephone1": "555000",
                "fax": "10101",
                "accountid": "60c4e274-0d87-e711-80e5-00155db19e6d"
            },
            {
                "@odata.context": "[Organization URI]/api/data/v9.1/$metadata#accounts/$deletedEntity",
                "id": "2e451703-c686-e711-80e5-00155db19e6d",
                "reason": "deleted"
            }
        ]
}
```

394

Alternate keys

- For inbound integration into Dataverse
- Create for existing column(s)
- Indexed
- Performance

```
using (var service = new OrganizationService(crmConnection))
{
    // Use alternate key (accountnumber) column to identify an account row
    Entity account = new Entity("account", "accountnumber", "MIC-12345");

    // Set new credit limit;
    account["creditlimit"] = new Money(100000);

    UpdateRequest request = new UpdateRequest() { Target = account };
    UpdateResponse response = (UpdateResponse)service.Execute(request);
}
```

© Copyright Microsoft Corporation. All rights reserved.

395

Upsert method

- Insert or Update in one request
- Use when don't know if row exists
- Need to provide "enough" information

```
using (var service = new OrganizationService(crmConnection))
{
    // Use alternate key (accountnumber) column to identify an account row
    var account = new Entity("account")
    {
        KeyAttributes = new KeyAttributeCollection
        {
            {"accountnumber", "MIC-12345" }
        }
    };

    account["name"] = "Microsoft";
    account["creditlimit"] = new Money(100000);

    var request = new UpsertRequest() { Target = account };
    var response = (UpsertResponse)service.Execute(request);
}
```

© Copyright Microsoft Corporation. All rights reserved.

396

 Microsoft Dynamics 365

© Copyright Microsoft Corporation. All rights reserved.

397

 Microsoft Power Platform

Learning Path 11: Custom Connectors

© Copyright Microsoft Corporation. All rights reserved.



398

Module 1: Get started with custom connectors



© Copyright Microsoft Corporation. All rights reserved.

399

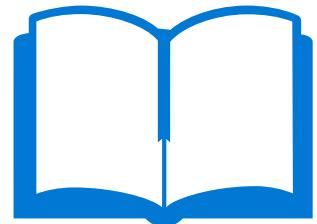
Microsoft Learn module

Get started with custom connectors in Power Automate

<https://learn.microsoft.com/training/modules/get-started-custom-connector>

Configure custom connectors with authenticated APIs in Power Automate

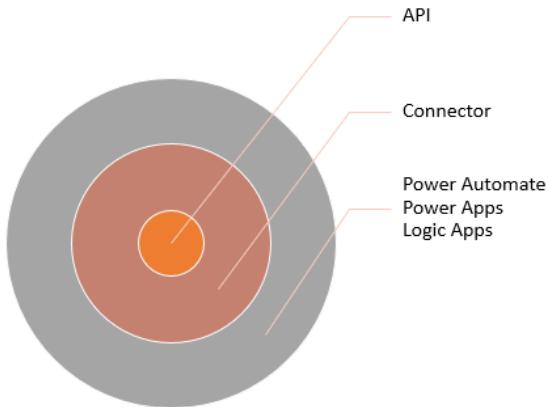
<https://learn.microsoft.com/training/modules/configure-custom-connectors-api>



© Copyright Microsoft Corporation. All rights reserved.

400

Introduction to custom connectors



© Copyright Microsoft Corporation. All rights reserved.

401

Custom connector lifecycle

A custom connector is a wrapper around a REST API that lets an underlying service communicate with Power Apps, Power Automate, or Azure Logic Apps.

Identify or build the API → Describe the API → Use the connector → Certify/ open source

© Copyright Microsoft Corporation. All rights reserved.

402

Identify the API

You can use any language and platform for your API, as long as it's made available as a REST API

© Copyright Microsoft Corporation. All rights reserved.

403

Build the API

Azure Functions can be used to create APIs that can be described by a custom connector

© Copyright Microsoft Corporation. All rights reserved.

404

Describe the API

You can create custom connectors using different approaches:

- Custom Connector wizard
- Import OpenAPI definition
- Import Postman collection
- Import from GitHub
- Create from Azure API Management

© Copyright Microsoft Corporation. All rights reserved.

405

Use the custom connector

You can use your custom connector the same way that you use any built-in connectors

© Copyright Microsoft Corporation. All rights reserved.

406

Share the custom connector

Connectors can be shared with other users in your organization the same way you share resources in Power Apps, Power Automate, and Azure Logic Apps.

© Copyright Microsoft Corporation. All rights reserved.

407

Configure API security

How a custom connector authenticates to the API



No authentication



Basic authentication – Username and password



OAuth 2.0 – Generic OAuth 2.0 or prebuilt configurations for Azure AD, GitHub, Slack, etc



API key – for example an Azure Function

© Copyright Microsoft Corporation. All rights reserved.

408

Create an Azure Function to be used in a custom connector

Create the Azure Function with the HTTP trigger

Authenticate using either

- API Key
- Azure AD

Create the custom connector either by:

- Creating manually with the custom connector wizard
- Adding the Azure Function to Azure API Management

409

© Copyright Microsoft Corporation. All rights reserved.

Module 2: Azure API Management and custom connectors



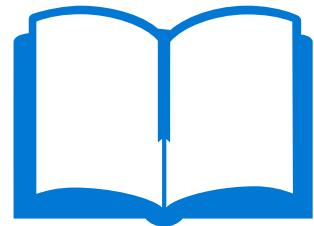
410

© Copyright Microsoft Corporation. All rights reserved.

Microsoft Learn module

Discover and use Web APIs with Power Apps

<https://learn.microsoft.com/training/modules/discover-web-apis-power-apps>



© Copyright Microsoft Corporation. All rights reserved.

411

Azure API Management

Azure API Management helps organizations publish web APIs to external, partner, and internal developers to unlock the potential of their data and services.

But before publishing the web APIs to be consumed by other developers, the author of the web APIs first must deploy them to Azure API Management.

swagger <http://localhost:60201/swagger/v1/swagger.json> My API V1 ▾

My API

Todo

		Show/Hide	List Operations	Expand Operations
GET	/api/Todo			
POST	/api/Todo			
DELETE	/api/Todo/{id}			
GET	/api/Todo/{id}			
PUT	/api/Todo/{id}			

[BASE URL: /, API VERSION: v1]

© Copyright Microsoft Corporation. All rights reserved.

412

Create a custom connector from Azure API Management

- Select environment
- Enter connector name

© Copyright Microsoft Corporation. All rights reserved.

413

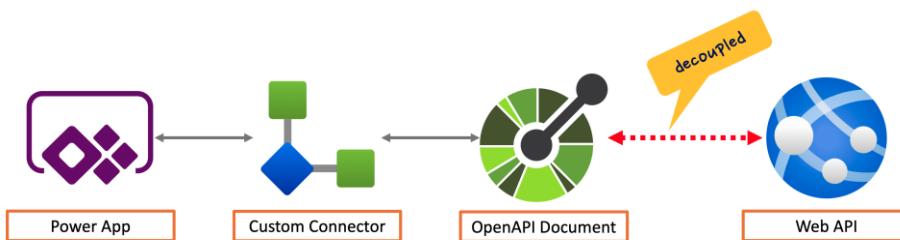
OpenAPI

OpenAPI defines a standard and programming language-agnostic interface description for web APIs.

It allows both humans and computers to discover and understand the capabilities of a service without having to access to source code, additional documentation, or inspect network traffic.

OpenAPI 2.0 (Swagger) JSON file

Power Platform lets you instantly create a custom connector with this OpenAPI document.

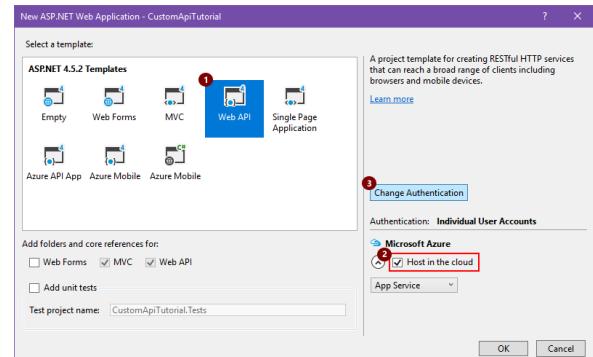


© Copyright Microsoft Corporation. All rights reserved.

414

Create a custom connector from ASP.NET Core web API

- ASP.NET web app in the Azure App Service
- Use Swashbuckle to create an OpenAPI file
- OpenAPI `http://<web-api-app-root-URL>/swagger/docs/v1`
- Import the OpenAPI to create custom connector



© Copyright Microsoft Corporation. All rights reserved.

415

OpenAPI vs Azure API Management

Connector	Azure API Management	OpenAPI
API Control	Centralized	Distributed
API Structure Layer	Complex	Simple
Extra Security Layer	Yes	No
Usage Control	Yes	No
Extra Cost	Yes	No
Architectural Complexity	Higher	Lower

© Copyright Microsoft Corporation. All rights reserved.

416

Module 3: OpenAPI definitions



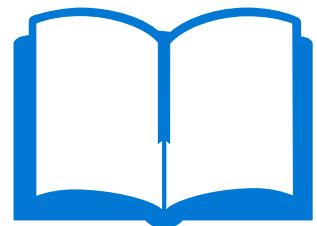
© Copyright Microsoft Corporation. All rights reserved.

417

Microsoft Learn module

Create Power Automate OpenAPI custom connectors

<https://learn.microsoft.com/training/modules/custom-connectors-open-api>



© Copyright Microsoft Corporation. All rights reserved.

418

OpenAPI extensions

The OpenAPI specification allows you to define extensions that augment the core definition to handle specific scenarios.

Microsoft has defined several extensions to the OpenAPI definition to allow you to configure specific features that are supported by custom connectors.

© Copyright Microsoft Corporation. All rights reserved.

419

OpenAPI extensions

The primary use case for configuring these extensions is to make the connector easier to use.

For example, without dynamic values, makers would have to know internal unfriendly schema values or strings for type/lookup parameters.

By configuring dynamic values, you're helping to provide a user-friendly drop-down list for makers to select from.



© Copyright Microsoft Corporation. All rights reserved.

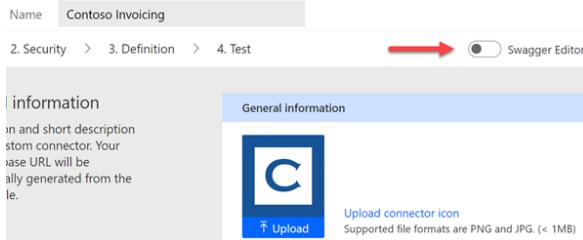
420

OpenAPI extensions

Identify extensions by looking for

`x-ms-<name>` in the OpenAPI definition

Add extensions using the Swagger editor



Example extensions

- `x-ms-capabilities`
- `x-ms-url-encoding`
- `x-ms-dynamic-values`
- `x-ms-dynamic-list`
- `x-ms-dynamic-schema`
- `x-ms-visibility`

421

Use OpenAPI extensions

Chunking is used to process large messages in an API

If the API supports chunking, then need to enable `chunkTransfer` in the OpenAPI definition

```
"x-ms-capabilities": {
    "chunkTransfer": true
}
```

© Copyright Microsoft Corporation. All rights reserved.

422

Module 4: Policy templates



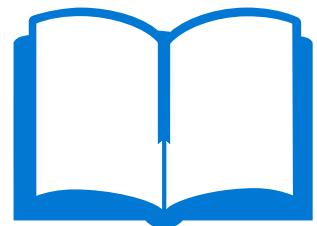
© Copyright Microsoft Corporation. All rights reserved.

423

Microsoft Learn module

Configure policy templates for custom connectors in Power Automate

<https://learn.microsoft.com/training/modules/policy-templates-custom-connectors>



© Copyright Microsoft Corporation. All rights reserved.

424

Policy templates

Policies allow you to modify the behavior of a custom connector at runtime.

You can use policies to perform data conversion, route requests, set parameter values, and more.

You can configure policies directly in the custom connector API properties file before import, or you can do it from the maker portal in the custom connector designer by applying policy templates.

© Copyright Microsoft Corporation. All rights reserved.

425

Use cases for policy templates

Route request

- Routes requests to a specified endpoint on the same service

Set Host URL

- Route calls to different endpoints
- Use expressions to access runtime values

Set HTTP header

- Override or add information to a request or response
- Enable CORS for API Management connectors

Set Query String Parameter

- Adds or updates value of request query string parameter

The screenshot shows the 'Policy details' section of the Azure portal. At the top, there's a blue header bar with the text 'Policy details'. Below it, there's a form with a 'Name' field containing the value 'Set Host Url'. Underneath the name field is a 'Template' dropdown menu. The menu has a heading 'Choose a template' and a list of items: 'Convert an array to an object (Preview)', 'Convert an object to an array (Preview)', 'Set host URL', 'Route request', 'Set connection status to unauthenticated (Preview)', 'Set HTTP header', 'Set property (Preview)', 'Set query string parameter', 'Set header/query parameter value from URL (Preview)', and 'Convert delimited string into array of objects (Preview)'. The 'Set host URL' option is currently selected, indicated by a grey background and a small orange dot next to the arrow icon.

© Copyright Microsoft Corporation. All rights reserved.

426

Learning Path 11 practice labs

Custom Connectors

Lab 13: Custom Connector
(Optional)



© Copyright Microsoft Corporation. All rights reserved.

427



© Copyright Microsoft Corporation. All rights reserved.

428



Learning Path 12: Application lifecycle management (ALM)



© Copyright Microsoft Corporation. All rights reserved.

429

Module 1: Solutions and Application Lifecycle Management



© Copyright Microsoft Corporation. All rights reserved.

430

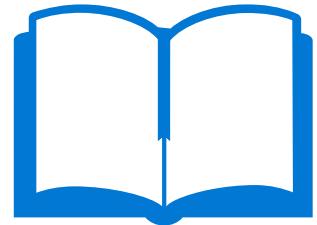
Microsoft Learn modules

Manage solutions in Power Apps and Power Automate

<https://learn.microsoft.com/training/modules/manage-solutions-power-automate>

Introduction to solutions for Microsoft Power Platform

<https://learn.microsoft.com/training/modules/introduction-solutions>

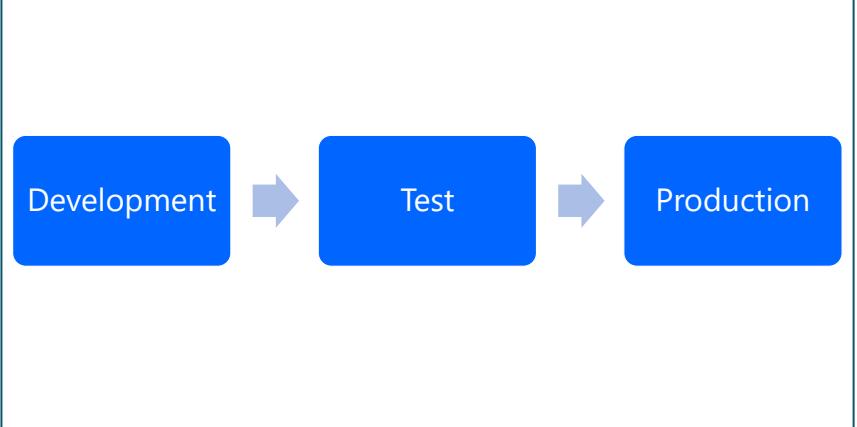


© Copyright Microsoft Corporation. All rights reserved.

431

Solutions

- Solutions are containers of Power Platform components
- Transport customizations from one environment to another
- Part of an overall strategy for application lifecycle management (ALM)



© Copyright Microsoft Corporation. All rights reserved.

432

Example solution

Solution – Device Ordering

Apps	Data (Tables)	Processes
Device Ordering	Device Order	Order Approval Flow
Device Procurement		Device Procurement Process
Security Roles		
Device Ordering App role		
Device Procurement App role		
Device Procurement Flow role		

Solutions > Contoso Device Order Management

Display name	Name	Type
Calculate Ship Date	contoso_calculateShipDate	Process
Contoso CDS Model Driven User	contoso_cdsModelDrivenUser	Security Rule
Contoso Device Order App	contoso_DeviceOrderApp	Security Rule
Contoso Device Procurement App	contoso_DeviceProcurementApp	Security Rule
Contoso Device Procurement Flow SP	contoso_DeviceProcurementFlowSP	Security Rule
Approval Status	contoso_ApprovalStatus	Global Set
Device Order	contoso_DeviceOrder	Activity
Device Ordering App	contoso_DeviceOrderingApp	Custom App
Device Procurement	contoso_DeviceProcurement	Model-driven App
Device Procurement	contoso_DeviceProcurement	Global Extension
Device Procurement Process	contoso_DeviceProcurementProcess	Activity
Device approval request	contoso_DeviceApprovalRequest	Process
Device Procurement Process	contoso_DeviceProcurementProcess	Process

© Copyright Microsoft Corporation. All rights reserved.

433

Types of solution



Unmanaged

- Used during development
- Used to transport to other development environments



Managed

- Used to distribute to non-development environments

© Copyright Microsoft Corporation. All rights reserved.

434

Default solutions



Common Data Service Default solution



Default solution

© Copyright Microsoft Corporation. All rights reserved.

435

Solution components



Solution components are categorized to track the individual items associated with the solution



Component types include tables, apps (model and canvas), Power Automate flows, security components as well as developer assets



Some components are sub-components, for example forms, views, charts, and columns are all assets of tables

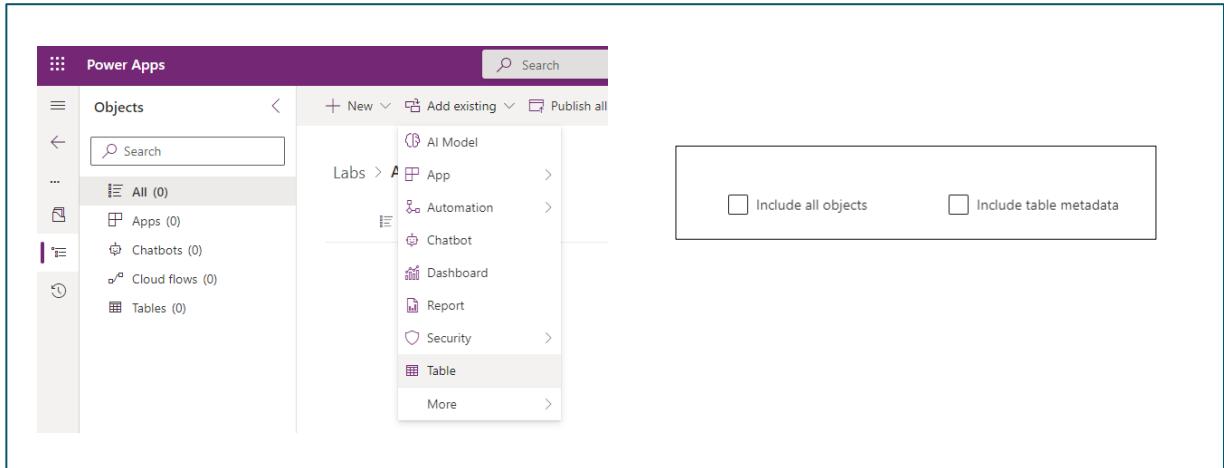


Reference data is not included as a solution component

© Copyright Microsoft Corporation. All rights reserved.

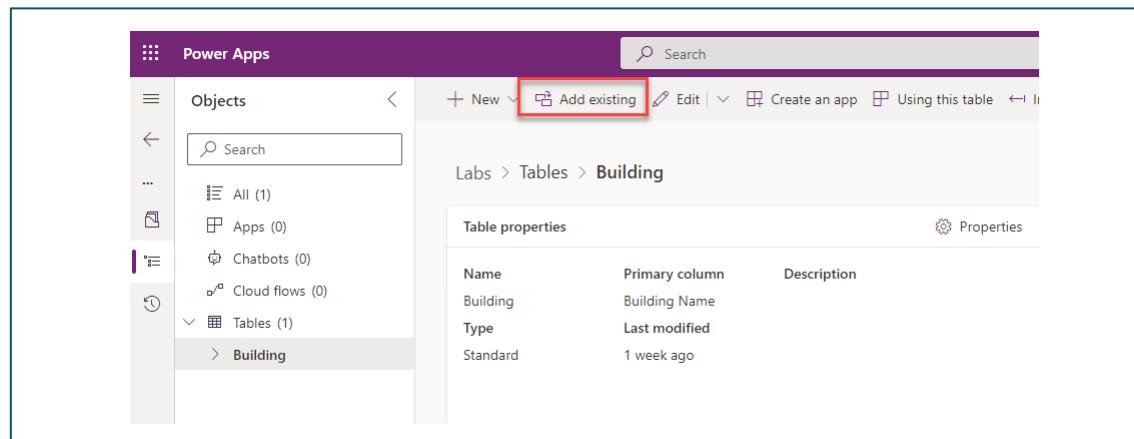
436

Segmented solutions



437

Add table assets



438

Removing Items

Labs > All

Display name ↑	Name ↓	Type ↓	Managed ↓	Last Modif... ↓	Or
Building	dem_building	Table	No	1 week ago	-

Remove from this solution
Delete from this environment

© Copyright Microsoft Corporation. All rights reserved.

439

Solution dependencies

Solution components can depend on other objects
Solution components can be required by other objects

Labs > All

Display name ↑	Name ↓	Type ↓
Building	dem_building	Table

Open | >
Edit | >
Import | >
Export | >
Properties
Publish
Advanced | > + Add required objects
Remove | > Show dependencies
See solution layers
Managed properties
Tools | >

© Copyright Microsoft Corporation. All rights reserved.

440

Environment variables



Apps and flows often require different configuration settings across environments e.g., Sender email, Portal URL etc.



Flows become configurable without hard coded values that must be edited upon deployment



Supports default values that can be overridden in each environment



Environment variables are solution components and participate in full application lifecycle management

© Copyright Microsoft Corporation. All rights reserved.

441

Creating environment variables

- Decimal number, Text, JSON, Yes/No or a secret
- Current value overrides any default value provided
- Remove the current value if you don't want it transported to the next level environment

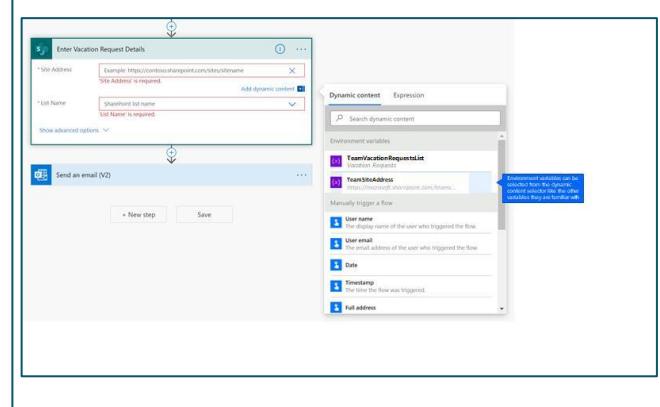
The screenshot shows the 'New environment variable' dialog box. It includes fields for 'Display name' (L1 Approver), 'Name' (contoso__L1Approver), 'Description' (L1 approver email), 'Data type' (Text), 'Default Value' (L1approvers@contoso.com), and a 'Current Value' section with a 'New value' input field. At the bottom are 'Save' and 'Cancel' buttons.

© Copyright Microsoft Corporation. All rights reserved.

442

Using environment variables

- Use the dynamic content panel to select variable values
- No need to manually query
- Use environment variable values in trigger properties
- Does not require a Microsoft Dataverse license to use variables

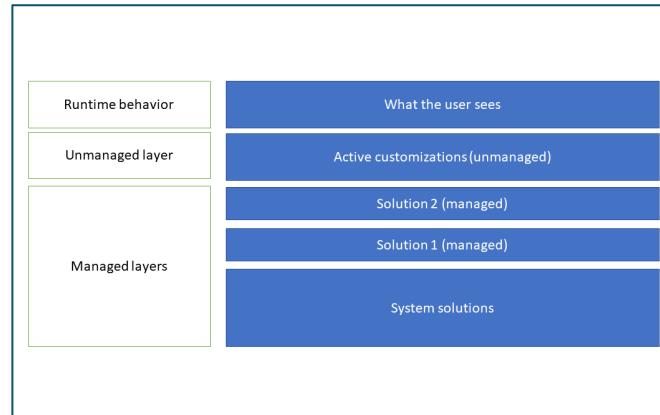


© Copyright Microsoft Corporation. All rights reserved.

443

Solution layers

- Managed layer(s)
- Unmanaged (Active) layer
- Editing or even just testing a managed flow without saving creates an unmanaged layer
- Delete active layer



© Copyright Microsoft Corporation. All rights reserved.

444

Solution checker



As Power Automate and Power Apps are used to customize a deployment, each offers their own inline app checkers that are helpful for real-time issue resolution.



Solution checker is able to look at the whole solution, do static analysis, and produce a detailed list of any issues found.



Solution checker should be run regularly on any unmanaged solution you are building in your development environments.

© Copyright Microsoft Corporation. All rights reserved.

445

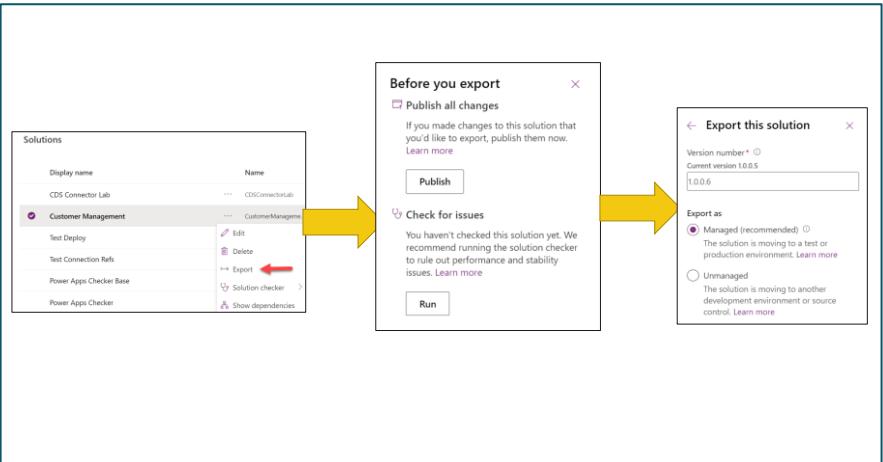
Manual export

Always export both managed and unmanaged

Managed solutions are deployed read-only to test/production

Unmanaged solutions can be used to restore your dev environment or store in source control

Publish all and check for issues should be part of your export process



© Copyright Microsoft Corporation. All rights reserved.

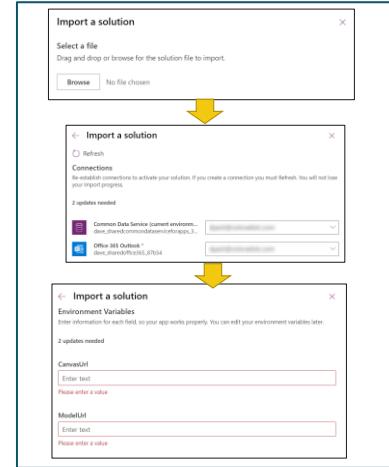
446

Manual import

Allows hooking up each connection reference to a valid connection or setting up new connections as required

Allows setting any environment variables

Connections and environment variable values are retained during updates



© Copyright Microsoft Corporation. All rights reserved.

447

Importing a new version of a managed solution



Upgrade – Default option, imports changes and applies them immediately including removing components not in new solution, old solution and any patches are removed. Recommended.



Stage for Upgrade – Similar to upgrade but it pauses after new solution is imported before removing components so you can do data migration etc. and then manually apply the solution.



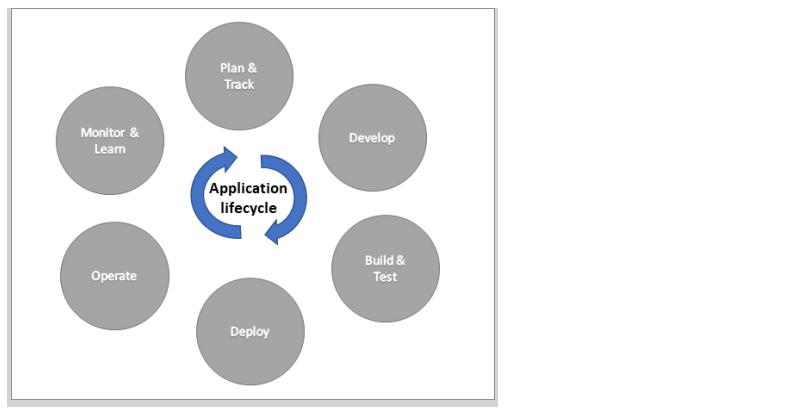
Update – Applies changes, no removal of items occur. Not recommended.

© Copyright Microsoft Corporation. All rights reserved.

448

Automate solution management

So far in this learning path you have seen how solutions can be used to track your app components and transport them from one environment to another manually.



© Copyright Microsoft Corporation. All rights reserved.

449

Automating ALM with DevOps tools



Source control is a feature of Azure DevOps and GitHub that allows you to keep track of versions of your solution components.



Automation is important because it brings consistency to what would otherwise be a manual process.

© Copyright Microsoft Corporation. All rights reserved.

450

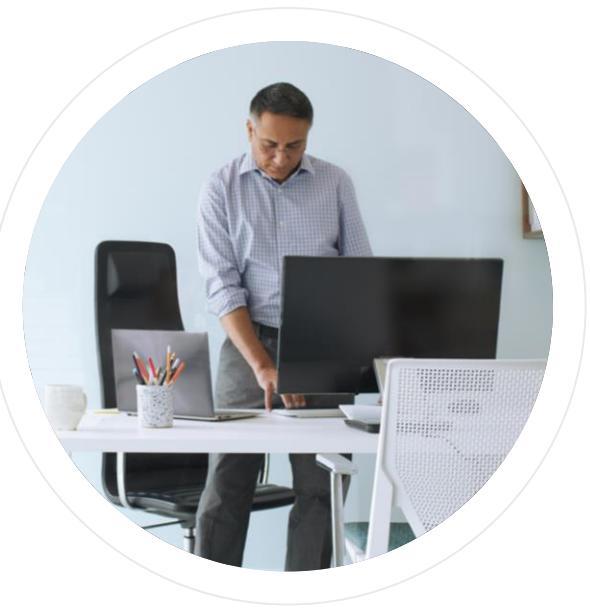


© Copyright Microsoft Corporation. All rights reserved.

451



Learning Path 13: Create a Technical Design



© Copyright Microsoft Corporation. All rights reserved.

452

Module 1: Technical architecture



© Copyright Microsoft Corporation. All rights reserved.

453

Design authentication and authorization strategy

Authentication is the process or action of verifying the identity of a user or process.

Azure Active Directory (Azure AD) is used for authentication for accessing the Power Platform.

Users can authenticate using their Azure AD tenant credentials.

When building integrations with Dataverse you may need to register applications in Azure AD and then add S2S accounts to allow external systems to access Dataverse.

When accessing Azure services and external systems you will need to define how to authenticate against those systems. Typically, you will use OAuth 2.0

Authorization is the process or action of verifying whether an authenticated user is authorized to access the resources that are being provided.

Access to Power Platform environments is controlled by Azure AD security groups.

Access to resources within a Power Platform environment is controlled by a combination of Dataverse security roles and sharing of resources.

© Copyright Microsoft Corporation. All rights reserved.

454

Determine whether requirements can be met with out-of-the-box functionality

As a developer, you should approach apps on Microsoft Power Platform from the perspective that writing code for achieving desired business application functionality should be considered as an exception to no-code and low-code approaches.

You should use the capabilities of the Power Platform instead of writing code whenever possible, for example:

- Rollup columns
- Calculated columns
- Column-level security

© Copyright Microsoft Corporation. All rights reserved.

455

Determine whether requirements can be met with out-of-the-box functionality

Calculated columns

- Calculated on retrieve
- Can use columns on table and columns in many-to-one relationships

Rollup columns

- Requires a one-to-many relationship
- Recalculated every hour
- Max 100 per environment
- Max 10 per table
- Supports hierarchical relationships

© Copyright Microsoft Corporation. All rights reserved.

456

Determine whether requirements can be met with out-of-the-box functionality



Calculated and rollup column values are read-only



Calculated and rollup column values are not audited



Calculated and rollup columns do not trigger workflow, code, or Power Automate cloud flows



Calculated columns cannot contain a calculated column from another table that also contains another column from a different table (spanning three tables)

© Copyright Microsoft Corporation. All rights reserved.

457

Determine whether requirements can be met with out-of-the-box functionality

Alternatives to calculated columns

- Business rule
- Client scripting (JavaScript)
- Real-time classic workflow
- Dataverse plug-in
- Power Automate cloud flow

Alternatives to rollup columns

- Dataverse plug-in
- Power Automate cloud flow

© Copyright Microsoft Corporation. All rights reserved.

458

Determine whether requirements can be met with out-of-the-box functionality

Client scripting

- Client scripting is actioned immediately in the user interface when the user exits a field

Calculated columns

- Calculated columns are calculated on retrieve, so client changes won't show until data is refreshed

© Copyright Microsoft Corporation. All rights reserved.

459

Determine whether requirements can be met with out-of-the-box functionality

When evaluating requirements, you should be fully aware and leverage functional solutions over code-first solutions where possible.

Power Automate cloud flows offer and can exceed functionality traditionally available within Dataverse classic workflows and plug-ins for example you can use a Power Automate cloud flow to cascade data changes down a one-to-many relationship e.g., change of address from account to contacts

Business rules can achieve common requirements that negate the need for JavaScript in model-driven app forms

© Copyright Microsoft Corporation. All rights reserved.

460

Determine whether requirements can be met with out-of-the-box functionality

JavaScript	Business rules
<ul style="list-style-type: none"> Model-driven app form only Operates on columns, sections, and tabs Use Web API to access any table/column Use with command buttons Supports complex logic Supports OnSave event Supports Notifications Full Web API support 	<ul style="list-style-type: none"> Model-driven app form and Dataverse Operates on columns only Limited to columns on table/form Shows error next to field in the form Supports simple logic Supports OnLoad and OnChange events only Supports Recommendations

© Copyright Microsoft Corporation. All rights reserved.

461

Determine when to use Logic Apps versus Power Automate cloud flows

Logic Apps	Power Automate cloud flow
------------	---------------------------

Logic Apps are licensed on a pay-as-you-go basis (consumption) or as part of a service plan (standard)

Power Automate is licensed through Microsoft 365 and grants a quota of flow runs per month

Build in Azure portal or with Visual Studio

Build in Power Automate portal

Management, logging, monitoring and alerts, with Azure Monitor and Azure Application Insights

Approvals and Notifications supported

© Copyright Microsoft Corporation. All rights reserved.

462

Determine when to use serverless computing, plug-ins, or Power Automate

Capability	Power Automate cloud flow	Dataverse classic workflow	Dataverse plug-in
Synchronous or Asynchronous	Asynchronous	Either	Either
Access External Data	Yes, using connectors	No	Yes, using APIs
Maintenance	Makers	Business users	Developers
Can Run As	Current user or flow owner	Current user or workflow owner	Any licensed user, system, or current user
Can Run On Demand	Yes	Yes	No
Can Nest Child Processes	Yes	Yes	Yes
Execution Stage	After	Before/After	Before/After
Triggers	Create, Column Change, Delete, On Demand, Scheduled	Create, Column Change, Delete, On Demand	Create, Column Change, Delete, plus many other messages

463

Determine when to use serverless computing, plug-ins, or Power Automate

Capability	Dataverse classic workflow	Dataverse plug-in	Client Scripting
Synchronous	Either	Either	Synchronous
Access External Data	No	Yes	Yes (with limitations)
Maintenance	Business Users	Developers	Developers
Can Run As	User	Any licensed user or current user	User
Can Run On Demand	Yes	No	No
Can Nest Child Processes	Yes	Yes	No
Execution Stage	Before/After	Before/After	Before/After
Triggers	Create, Column Change, Status Change, Assign to Owner, Delete, along with many other specialized triggers	Create, Column Change, Status Change, Assign to Owner, Delete, along with many other specialized triggers	Column Change or Form Load/Save

464

Determine when to use serverless computing, plug-ins, or Power Automate

Azure Functions

- Consumption plan 5-minute default and can increased to 10-minute
- Durable functions allow chaining of functions
- Premium and Dedicated plans have longer
- Need to Authenticate

Dataverse Plug-in

- Event driven
- Cannot schedule
- 2-minute limit
- Authenticated automatically

© Copyright Microsoft Corporation. All rights reserved.

465

Design Azure inbound and outbound integrations

Service Bus

- Asynchronous
- Register with Plugin Registration Tool
- Invoke with no code or with code
- Requests can be queued

Webhook

- Synchronous or Asynchronous
- Register with Plugin Registration Tool
- Invoke with no code or with code
- Requests dependent on HTTP endpoint being available

© Copyright Microsoft Corporation. All rights reserved.

466



© Copyright Microsoft Corporation. All rights reserved.