



# The Developer's 7-Step Guide to Low-Code App Development





# What is Power Apps?

Power Apps is a low-code app development platform that allows you to accelerate foundational app development so you can focus on custom components, new innovations, and testing cycles.

## Benefits

- ✓ Reduce development time and continuously iterate in real-time, without sacrificing functionality.
- ✓ Seamlessly integrate cloud-based and on-premises data using prebuilt connectors or custom APIs.
- ✓ Incorporate AI with prebuilt components to add powerful in-app intelligence.

Power Apps leverages the secure, built-in Microsoft Dataverse that allows you to store data and model processes in a standard format. Power Apps also natively integrates with Microsoft productivity tools such as Office 365, Azure, Dynamics 365, and other solutions within the Microsoft Power Platform.

Using Power Apps, you can build apps by leveraging the following:

- ✓ [Canvas components](#) allow you to design and build a fully customized app without a template.
- ✓ [Model-driven components](#) allow you to build an app with the layout determined by the components you add.
- ✓ [Portals](#) are external-facing websites that allow outside users to sign in.

The components and portals can be used as needed, even together in one app. For the purposes of this guide, we will begin building an app via the model-driven component functionality, which is often used to define and enforce consistent business processes.

## Prerequisites

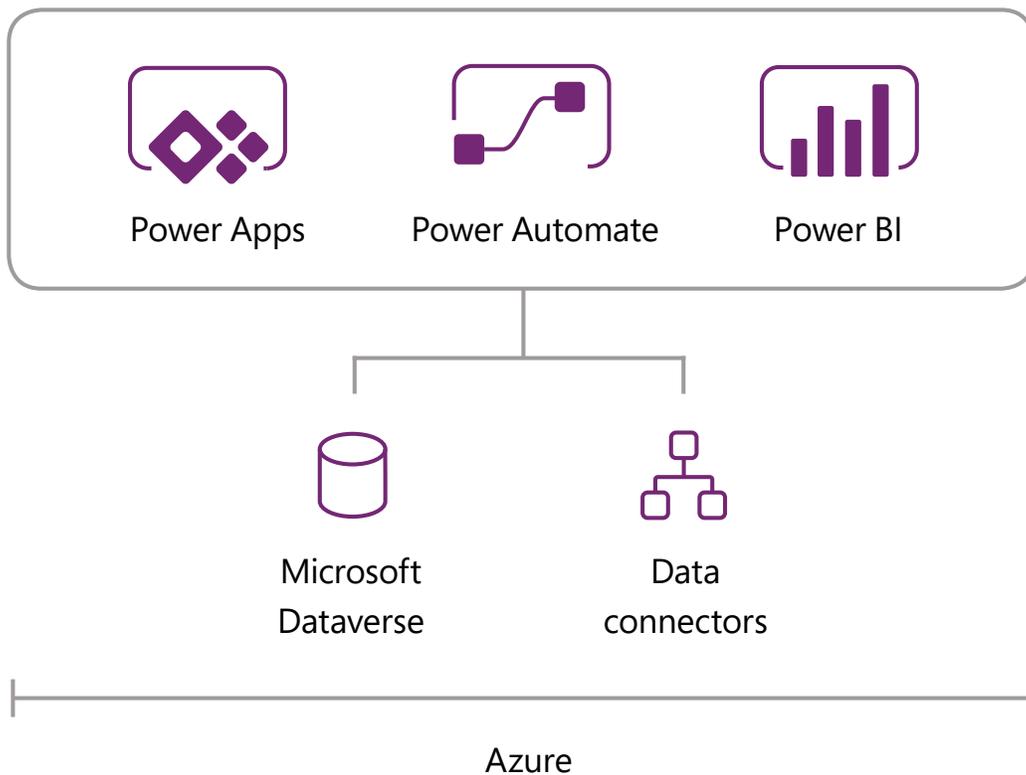
Complete these prerequisites to get started:

- Sign up for a [free 30-day trial](#) of Power Apps or a [community plan](#)
- Understand [model-driven app components](#)

STEP  
**1**

# Define data for your app

Define your app data in Microsoft Dataverse using the following components: entity, field, relationship.



## Entities

[Entities](#) are used to model and manage business data. You can use standard entities, custom entities, or both.

### How to create an entity

- ✓ In the navigation pane of Power Apps, select Data and then select **Entities**.
- ✓ In the command bar, select **New entity**.
- ✓ In the New entity panel, do the following:
  1. In the Display name box, enter **Product review**.
  2. In the **Primary Field** section, in the **Display name box**, replace **Name** with **Product review**.
  3. Open the **More settings** section and expand the Description accordion. Enter a description (this is optional).
  4. From the **Choose entity type** drop-down list, select **Activity**.
  5. Ensure **Display in Activity menus** check box is selected.
  6. Expand **Create and update settings** and select the **Enable quick create forms** check box.
  7. Click **Create**.

- ✔ The entity is now being provisioned in the background. Once provisioning is completed, your entity will be saved and available for use in apps. Fields, relationships, and keys can be added to your entity at any time (even while provisioning is still in progress), but views, forms, charts, dashboards, and business rules can only be added to the entity after provisioning is completed.
- ✔ Under the **Fields** tab, observe the **Primary Field** that you named in the previous step. Select the **Primary Field** to open the **Primary Field** panel if you would like to make any additional customizations to the field. Notice that the **Name** can no longer be changed, since the entity has already been saved.
- ✔ To add a field to the entity, do the following:
  1. In the command bar, select **Add field** to open the **Field properties** panel.
  2. In the **Display name** box, enter **Review Date**.
  3. From the **Data type** drop-down list, select **Date Only**.
  4. Select the **Required** check box.
  5. Select **Done**.

- ✓ Repeat the previous step to add three more fields with the following configurations:  
**Display name** = Product Rating; **Data type** = Whole Number; click or tap **Required** check box.  
**Display name** = Reviewer Name; **Data type** = Text.  
**Display name** = Reviewer Comment; **Data type** = Text.
- ✓ Click **Save Entity** to save the latest changes to your entity.

## Fields

[Fields](#) define the individual data items that can be used to store information in an entity. You can create new fields to capture data if existing standard entities don't have fields that meet your needs.

### How to create fields for Microsoft Dataverse in Power Apps

- ✓ From the Power Apps portal, select **Data > Entities**, and select the entity that has the fields you want to view.
- ✓ While viewing fields, in the command bar, click **Add field**, which will show the **Field properties** panel.
- ✓ Set the **Display Name**, **Name**, and **Data type** properties, and click **Done** to close the **Field properties** panel.
- ✓ You can continue to edit the entity and add fields or return and continue editing this field. Fields are not created until you click **Save Entity** to save all the changes to the entity.

## Relationships

[Relationships](#) define how entity records can be associated with records from other entities or the same entity. There are two types of entity relationships: one-to-many and many-to-many.

### How to add a one-to-many relationship

To add a one-to-many relationship, create a relation under the **Relationships** tab and specify the entity with which you want to create a relationship.

- ✓ On powerapps.com, expand the **Data** section, and select **Entities** in the left navigation pane.
- ✓ Select an existing entity or create a new one.
- ✓ Click **Relationships**.
- ✓ Click the down arrow to the right of **Add relationship**. Click **One-to-many**, which will open a new panel to choose the entity you want to create a relationship to. Select the entity from the **Related entity** drop down.
- ✓ The **Look up** fields will be shown on the Primary entity. They will default with the entity's name, but you can change them if needed.
- ✓ Click **Done** to add the relationship to your entity and then click **Save entity**.

### How to add a many-to-many relationship

To add a many-to-many relationship, create a relation under the **Relationships** tab and specify the entity with which you want to create a relationship.

- ✓ On powerapps.com, expand the **Data** section, and select **Entities** in the left navigation pane.
- ✓ Select an existing entity or create a new entity.
- ✓ Click **Relationships**.
- ✓ Click the down arrow to the right of **Add relationship**. Click **Many-to-many**, which will open a new panel for you to choose the entity you want to create a relationship to. Select the entity from the **Related entity** drop down.
- ✓ The names for the relationship and relationship entity will appear. They will default to the names of the entities combined, but you can change them.
- ✓ Click Done to add the relationship to your entity, and then click **Save entity**.

## Building custom connectors

You can integrate data from a variety of services, such as GitHub and Salesforce, using pre-built connectors in Power Apps. You can also build custom connectors to call APIs, services, and systems that aren't available as pre-built connectors. There are three ways to create a custom connector:

- ✓ [From an OpenAPI definition](#)
- ✓ [From scratch](#)
- ✓ [From a Postman collection](#)

STEP  
 2

# Design the interface

Apps built in Power Apps are comprised of components such as entities, dashboards, forms, views, charts, and business processes. Use the [app designer](#) to bring these components together and define your site map, which defines the navigation for your app.

## Support matrix for the app designer and site map designer

The following table shows the supported operating systems and browsers.

| Browser/OS                  | Windows 10 | Windows 8.1 | Windows 8 | Mac OS X |
|-----------------------------|------------|-------------|-----------|----------|
| Microsoft Edge              | Yes        |             |           |          |
| Internet Explorer 11        | Yes        | Yes         |           |          |
| Internet Explorer 11 Modern |            | Yes         |           |          |
| Internet Explorer 10        |            |             | Yes       |          |
| Internet Explorer 10 Modern |            |             | Yes       |          |
| Mozilla Firefox             | Yes        | Yes         | Yes       |          |
| Google Chrome               | Yes        | Yes         | Yes       |          |
| Apple Safari                |            |             |           | Yes      |

You can use the [Power Apps component framework](#) to create code components that enhance the user experience when users are working with data on forms, views, and dashboards. For example, you can:

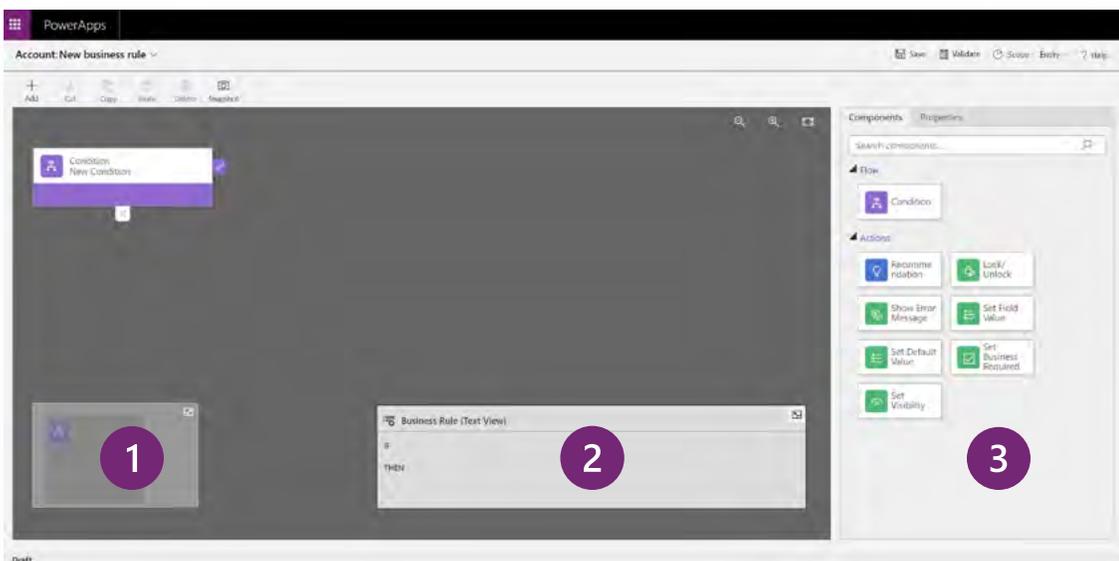
- ✓ Replace a field that displays a numeric text value with a dial or slider code component.
- ✓ Transform a list into an entirely different visual experience bound to the data set, such as a Calendar or Map.

STEP  
3Apply  
business logic

You don't have to write JavaScript or create plug-ins to [create business rules and apply form logic](#). By combining conditions and actions, you can do any of the following with business rules:

- ✓ Set field values
- ✓ Clear field values
- ✓ Set field requirement levels
- ✓ Show or hide fields
- ✓ Enable or disable fields
- ✓ Validate data and show error messages
- ✓ Create business recommendations based on business intelligence

The Business Rule designer window opens with a single condition already created for you. Every rule starts with a condition. The business rule takes one or more actions based on that condition.

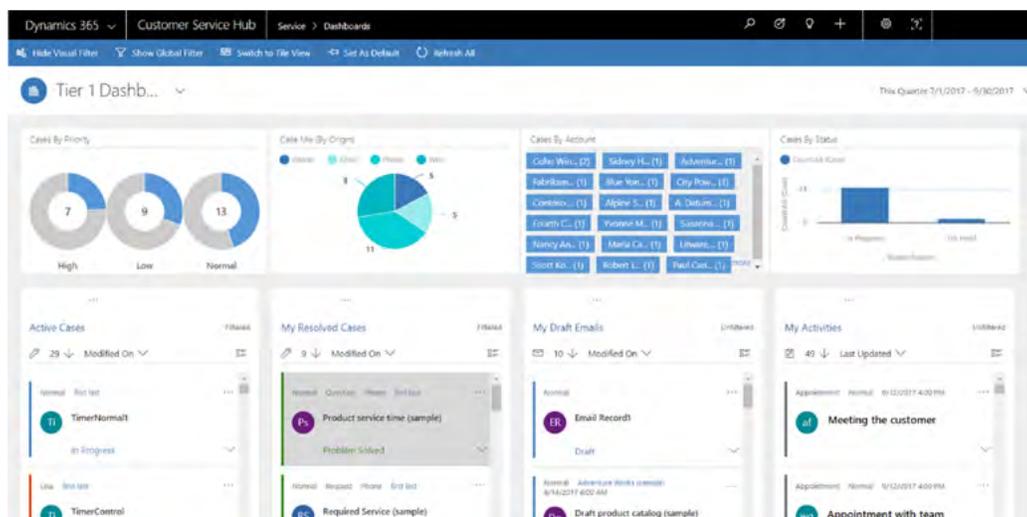


- 1 Use the mini-map to navigate quickly to part of the business rule.
- 2 As you create the business rule, Dynamic 365 builds the code here.
- 3 Drag components to the design area and set properties to create a business rule.

STEP  
4Apply data  
visualizations

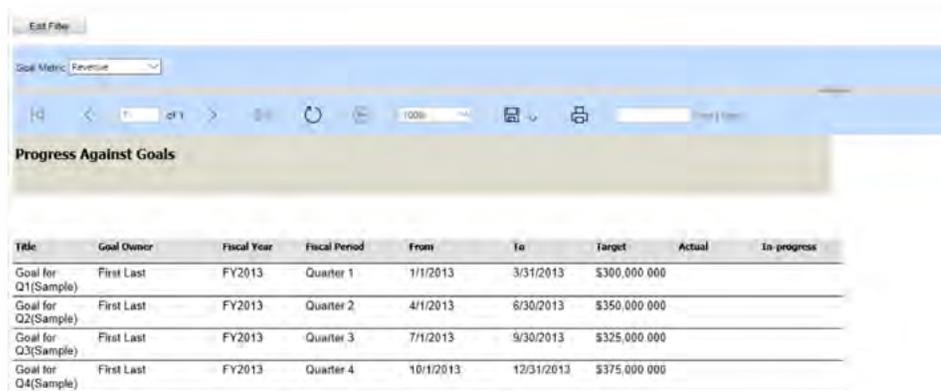
There are several ways to [apply data visualizations](#) to your app:

- ✓ [System charts](#) are organization-owned, so they are available to anyone with access to read the data running the app.
- ✓ [Dashboards](#) can be created by either an app user or a developer. As a developer, you can create and customize system dashboards that, when published, are visible to all app users. User dashboards are only visible to the user.
- ✓ [Interactive experience dashboards](#) are fully configurable, security-role based, and deliver workload information across multiple streams in real time.



# STEP 5 | Add reporting

Reports added to your app in Power Apps are based on SQL Server Reporting Services. They will therefore have the same set of features that are available for typical SQL Server Reporting Services reports. You can [add fetch-based reporting functionality](#) so that users can run, share, create, and edit reports, or you can create a new report by using the [Report Wizard](#) or the [Report Authoring Extension](#).



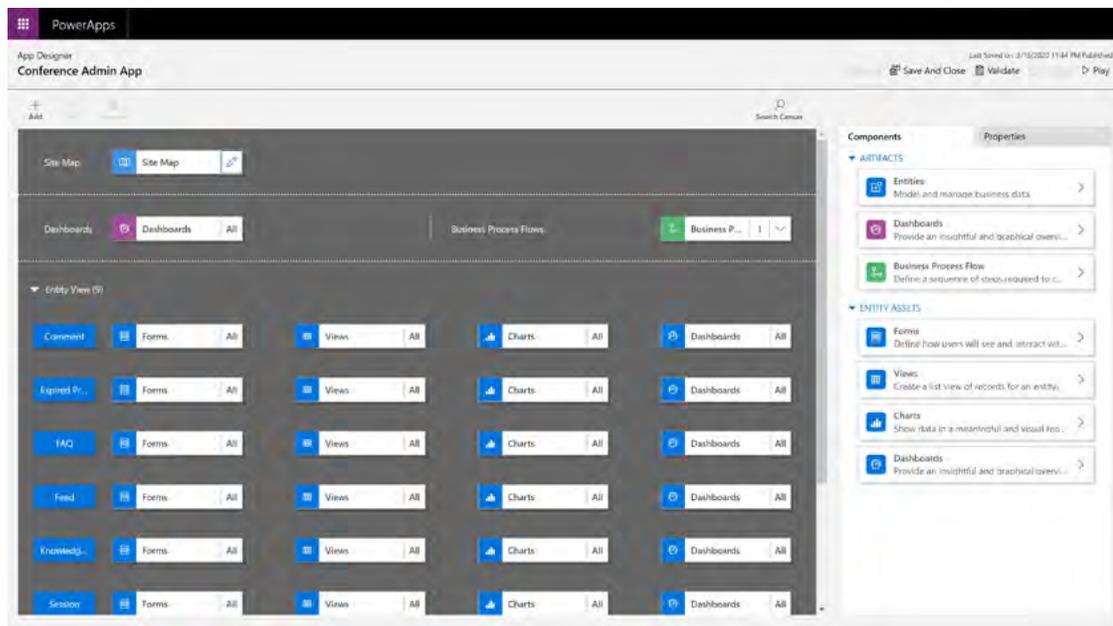
The screenshot shows a report interface with a table titled "Progress Against Goals". The table has columns for Title, Goal Owner, Fiscal Year, Fiscal Period, From, To, Target, Actual, and In-progress. The data is as follows:

| Title               | Goal Owner | Fiscal Year | Fiscal Period | From      | To         | Target        | Actual | In-progress |
|---------------------|------------|-------------|---------------|-----------|------------|---------------|--------|-------------|
| Goal for Q1(Sample) | First Last | FY2013      | Quarter 1     | 1/1/2013  | 3/31/2013  | \$300,000,000 |        |             |
| Goal for Q2(Sample) | First Last | FY2013      | Quarter 2     | 4/1/2013  | 6/30/2013  | \$350,000,000 |        |             |
| Goal for Q3(Sample) | First Last | FY2013      | Quarter 3     | 7/1/2013  | 9/30/2013  | \$325,000,000 |        |             |
| Goal for Q4(Sample) | First Last | FY2013      | Quarter 4     | 10/1/2013 | 12/31/2013 | \$375,000,000 |        |             |



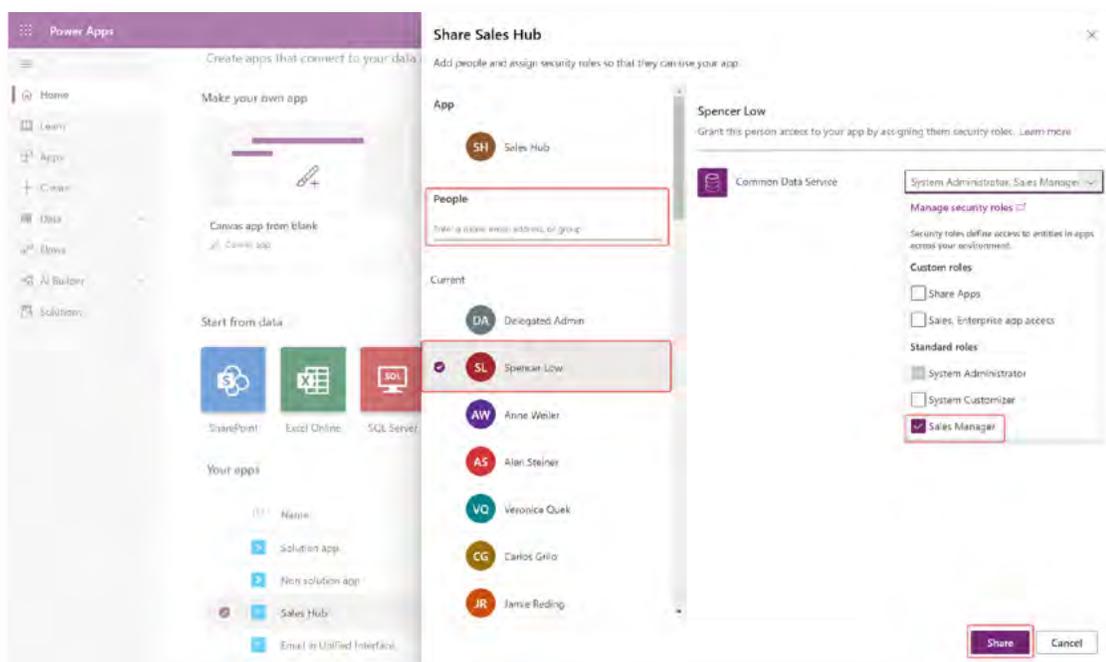
STEP  
6Validate and  
publish your app

Using the app designer, [validate your app](#) to check for asset dependencies that are required for it to work but haven't yet been added. After adding any missing dependencies and achieving a successful validation, use the app designer to [publish your app](#).



# STEP 7 | Share your app

Power Apps uses role-based security for sharing. [Security roles](#) must be assigned to your app, and anyone using the app must be assigned to one or more predefined or custom roles. Or, roles can also be assigned to teams.



You can share your app for [basic use](#) or a [specific use](#). You can also use [Azure Active Directory Groups](#) to manage access.



# Conclusion

You have now used model-driven components to create an app in Power Apps—and that’s just the beginning. Next, learn how to [manage your app](#) or explore [advanced app-making and customization](#).

Keep up to date on new features that have been recently released and upcoming releases.

[Keep up](#) >

Watch expert tutorials on how to leverage the features and functions of Microsoft Power Apps.

[Learn more](#) >

Get tips and updates on using Power Apps from the experts at Microsoft.

[Read more](#) >

Connect with peers, share ideas, and learn from experts in the Power Apps Community.

[Join us](#) >

©2021 Microsoft Corporation. All rights reserved. This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it. This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.