Introduction

The following Getting Started Guide is designed to walk ISVs new to the Dynamics 365 platform through the necessary steps to get started connecting their external apps with Dynamics 365 business applications.

This guide is broken down into five sections:

**Becoming a Microsoft partner**
First, it outlines the process to become a Microsoft partner, including registering for the Microsoft Partner Network (MPN) and Partner Center (PC). If you are already a Microsoft partner with PC access, you can skip this section.

**Understanding connectors**
The second section provides an overview of connectors to help you understand what connectors are and how they work, including a look at the Power Platform to provide context and clarity around the structure and available tools.

**How to connect**
The third section walks through how to create and use connectors using the Power Platform.

**Publishing a connector**
The fourth section outlines the ways to publish a connector, whether sharing internally, through the connector store on the Microsoft Flow website, or through an app integration on AppSource.

**Resources**
Lastly, we’ve included links to additional resources to help you find answers to questions and provide additional guidance for when you’re ready to move beyond the basics.

Let’s first take a look at the three programming models and define what Connect means.
Introduction

Programming models
While our Business Applications Platform is made up of best of breed applications—including Finance and Operations, CRM, Office 365, and LinkedIn—the true power of the platform is in how it all works together through our Common Data Service (CDS).

The CDS is the shared data language used by business and analytical applications. It consists of a set of a standardized, extensible data schemas published by Microsoft and our partners that enables consistency of data and its meaning across applications and business processes.

This means that when you’re interacting with your data—whether reading or writing—you are doing so from a common data source. There is no more tying together disparate systems and retrofitting data connections. It all works seamlessly through the CDS.

This model unlocks three key opportunities for ISVs who wish to leverage our Business Applications Platform for their business: Build, Extend, and Connect.

Build
Build standalone apps on the Common Data Service
ISVs can build standalone business apps directly on the CDS using the Power Platform (i.e., PowerApps, Power BI, and Microsoft Flow).

Extend
Extend the functionality of Dynamics 365 business applications
ISVs can extend the functionality of a Dynamics 365 business applications, such as Dynamics 365 for Sales and Dynamics 365 for Finance and Operations. Extend scenarios include creating industry or vertical customizations to our first-party apps.

Connect
Connect to first-party apps from external apps
ISVs can connect their external solutions to first-party Dynamics 365 apps or the Power Platform.

Licensing requirements
Build
PowerApps Plan 2 is the minimum required plan. Refer PowerApps pricing for more details.

Extend
Full PowerApps Plan 2 with a license for the required Dynamics 365 application is needed. Refer PowerApps pricing and PowerApps for Dynamics 365 for more details.

Connect
PowerApps Plan 1 is the minimum required plan. Refer PowerApps pricing for more details.
Contents

5 Prerequisites
   Sign up for Power Platform
   Become a Microsoft partner
   Register as a publisher

8 Understanding connectors
   Parts of the Power Platform
   Understanding connectors
   Using connectors

17 Building a custom connector
   Build and secure your API
   Describe the API and define the custom connector
   Securing your custom connector
   Testing your custom connector

22 Publish a custom connector
   Share your connector internally
   Publish to Connector store
   Publish to AppSource

33 Additional resources
Prerequisites

Sign up for Power Platform
• Register for a PowerApps account

Step 1
Become a Microsoft partner
• Create a Microsoft work account
• Join the Microsoft Partner Network
• Set up your Partner Center account

Step 2
Register as a publisher
• Register in Partner Center
• Sign up for the Dynamics Insider Program

If you are already a Microsoft Partner with Partner Center access, you can skip this section.
Prerequisites

Before you publish, market, and sell your first apps, there are a few steps you must take to get set up as a Microsoft Partner. These include joining the Microsoft Partner Network (MPN) and getting registered in the programs that enable you to publish, market, and sell your apps.

Sign up for Power Platform

Register for a Power Platform account
To begin building PowerApps, you must set up an account. You can try PowerApps for free by signing up either for a 30-day trial or a community plan.

1 Become a partner

Becoming a Microsoft partner gives you access to the Microsoft resources needed to build, market, and sell your apps. While you don’t need to be a Microsoft partner to begin developing your apps, all of the steps below are required to gain access to the programs that enable you to publish, market, and sell your apps.

Obtain a work account
Your work account or work email is the email address provided to you by your company. A work account email is usually in the format you@yourcompany.com. More information on work accounts can be found here.

Join the Microsoft Partner Network
Becoming a Microsoft partner gives you access to all the resources you need to build and publish apps. To become a partner, you must join the Microsoft Partner Network (MPN), at which time you will be assigned an MPN ID. MPN membership is free to all partners; you can enroll in the MPN here.

To build and submit an API connector, your app must fit the following criteria:
- Business user scenario that fits well with Flow, PowerApps and Logic Apps
- Publicly available service with stable REST APIs

If you have an active subscription to Microsoft Azure or Office 365, you already have a Microsoft work account.

Sign in for an enhanced partner experience

Sign in to Microsoft Partner Network (MPN) for a more personalized experience. Not a partner yet? Find out where partnership with Microsoft can take you.
Prerequisites

**Set up your Partner Center account**
Once you have joined the Microsoft Partner Network (MPN), you can set up your Partner Center (PC) account. Your PC account provides you with access to pricing information, tools and services, and enables you to manage admin credentials for your company's work account. PC is also where you can purchase or renew subscriptions to Microsoft Action Packs, create a business profile to receive and manage sales leads from Microsoft, and see if you qualify for co-selling opportunities.

2 **Register as a publisher**
Registering as a publisher allows you to sell your solutions on AppSource, the marketplace that gives ISVs access to more than 100 million commercial users.

**Register in Partner Center**
The first step to becoming a publisher is to register in Partner Center (PC). PC is where you submit your apps for publication, promote your apps, and manage your offers.

To begin the registration process, you must complete these steps. Shortly thereafter, one of our team members will follow up to help you complete your registration.

Once registered, you can access PC.

**Sign up for the Dynamics Insider Program**
In addition to the above steps, all partners must join the Dynamics Insider Program. Through the Dynamics Insider Program, you can gain access to partner resources, test and validate new features, and provide your valuable feedback.
# Understanding connectors

## Parts of the Power Platform
- PowerApps
- Microsoft Flow
- Power BI
- Common Data Service

## Understanding connectors
- What is a connector?
- What can connectors do?
- Components of a connector
- Public connectors vs. Custom connectors
- Why build a connector?
- Custom connector architecture

## Using connectors
- Using connectors within multiple products
  - PowerApps
  - Microsoft Flow
  - Logic Apps
Parts of the Power Platform

Power Platform enables business to create custom solutions—from simple to complex—through familiar, intuitive technology. Power Platform enables faster data collection, surfaces real-time insights, and empowers users to make informed, actionable decisions. It also enables users to do three key actions on data that help them drive business: gain insights from data (Analyze), drive intelligent business processes (Act) via apps they build, and automate the processes (Automate). Analyze, Act, and Automate—all done with Common Data Service (CDS), Power BI, PowerApps, and Flow, all working together atop data to help everyone.

Microsoft’s Power Platform is comprised of three tools—PowerApps, Microsoft Flow, and Power BI—designed to help you build, extend, and connect apps for Dynamics 365 and Office 365.

PowerApps

PowerApps sits at the center of the Power Platform. PowerApps is a collection of services, apps, and connectors that work together to let you build applications, ranging from simple no-code mobile apps used to view and update your data to fully featured software that extends the functionality of our Dynamics 365 solutions. With PowerApps, you do much more than just view your data. You can act on your data and update it anywhere and from any device.

Sign up for a free trial version and sign in and explore samples and templates.

Building blocks

To create, share, and administer apps, you’ll use these sites:

1. make.powerapps.com: In the web portal, you can open apps, specify the type of app that you want to create, and create data connections and flows. To use this site, you’ll need to log in using your organizational account.

3. **PowerApps Mobile**: With PowerApps Mobile, you can run your apps on Microsoft Windows, Apple iOS, and Google Android devices.

4. **PowerApps admin center**: In the admin center, you can define environments and data policies.

### Related technologies, platforms, and data sources
Microsoft PowerApps works with other technologies, platforms, and data sources to help you build and share apps across your organization. Let's take a look at some of these:

- **Microsoft Dynamics 365**: Dynamics 365 is the home for all your business apps, including Dynamics 365 for Sales, Dynamics 365 for Finance and Operations, Dynamics 365 for Field Service, and many others.

- **Microsoft AppSource**: AppSource is the marketplace through which you share your apps and download other apps that might be useful to your business.

- **Data sources**: Data sources bring cloud and on-premises data into your apps. You access data through built-in connections, custom connectors, and gateways.

### Microsoft Flow
With Microsoft Flow, users can connect to more than 200 services to create automated, multi-step workflows, as well as easily create custom connections when needed. Once built in Flow, these workflows and connections can be leveraged to extend the functionality of your PowerApps solution. Sign in here and explore templates.

### Power BI
Power BI enables users to build sophisticated, visual dashboards from their data. As part of the Power Platform, Power BI dashboards components can easily be embedded into PowerApps solution, and PowerApps components can be embedded into Power BI dashboards. Sign in here and explore templates.

### Build and certify custom connectors
Bring your product into the Microsoft cloud by building a connector that enables PowerApps to talk to your service.

### Common Data Service
Microsoft’s Business Applications—including Dynamics 365, Office 365, and Power Platform—are all built on top of our Common Data Service (CDS). CDS for Apps lets you securely store and manage data used by business applications. Data within CDS is stored within a set of records called entities. An entity is a set of records used to store data, similar to how a table stores data within a database.
CDS includes a base set of standard entities that support common business scenarios that connect to Dynamics 365 application data. You can also create custom entities specific to your organization and populate them with data that you import from lists in SharePoint, Excel, or PowerQuery. App makers can then use PowerApps to build rich applications using this data.

Dynamics 365 applications, including Dynamics 365 for Sales, Field Service, Customer Service, and Talent, use CDS for Apps to store and secure data used by the applications. This means you can build apps using PowerApps and CDS for Apps directly against your core business data already used within Dynamics 365 without the need for manual integration. Dynamics 365 for Finance and Operations and Dynamics 365 for Retail currently require configuration of the Data Integrator to make your business data available within CDS for Apps.

Understanding connectors

What is a connector?
A connector is a proxy or a wrapper around an API that allows the underlying service to talk to Microsoft Flow, PowerApps, and Logic Apps. It provides a way for users to connect their accounts and leverage a set of pre-built actions and triggers to build their apps and workflows.

Our large ecosystem of software as a service (SaaS) connectors enable you to connect apps, data, and devices in the cloud. Examples of popular connectors include Salesforce, Office 365, Twitter, Dropbox, Google services, and more. We currently have over 200 connectors.

Using custom connectors, you can connect custom APIs or to data in your local systems through the common on premise data gateway. Your apps will be able to fully interact with underlying sources and will respect the permissions you’ve established with them.
What can a connector do?
Developing a single connector enables an integration for three different Microsoft products: PowerApps, Microsoft Flow, and Logic Apps.

PowerApps
PowerApps enables users to build cloud-connected and cross-platform business apps using clicks and minimal/no code. Using PowerApps, your customers can leverage connectors to build simple apps for line of business scenarios that read and write data to multiple cloud sources. Some examples of such apps include survey forms, timesheets, expense reporting, etc. Users can securely publish and share these apps to the web or mobile for use within their organization.

Microsoft Flow
Using Microsoft Flow, your customers can automate tasks and build workflows in conjunction with other social and business applications. The possible workflows span across a wide variety of possibilities, for example:

- Send email, text, and push notifications.
- Copy files between data sources.
- Automatically collect and organize business data.
- Streamline approvals and send instant alerts.

Logic Apps
Logic Apps is the workflow engine that powers Microsoft Flow. It enables pro-developers to visually or programmatically configure workflows in Azure. A connector in Logic Apps can enable your customers to automate EAI, business to business (B2B), and business to consumer (B2C) scenarios, while reaping the benefits of source control, testing, support, and operations.

Components of a connector
Each connector offers a set of operations, classified as Triggers and Actions. Once you connect to the underlying service, these operations can be easily leveraged within your apps and workflows.

Triggers
Several connectors provide triggers that can notify your app when specific events occur. For example, the FTP connector has the OnUpdatedFile trigger. You can build either a Logic App or Flow that listens to this trigger and performs an action whenever the trigger fires.
Understanding connectors

There are two types of trigger.

- **Polling Triggers**: These triggers call your service at a specified frequency to check for new data. When new data is available, it causes a new run of your workflow instance with the data as input.

- **Push Triggers**: These triggers listen for data on an endpoint, that is, they wait for an event to occur. The occurrence of this event causes a new run of your workflow instance.

**Actions**

Actions are changes directed by a user. For example, you would use an action to look up, write, update, or delete data in a SQL database. All actions directly map to operations defined in the OpenAPI definition.

**Public connectors vs. Custom connectors**

**Public connectors**

Public connectors are available to all users. With a growing collection of over 200 public connectors currently available, these connectors allow you to easily create automated workflows between your favorite apps and services, including Office 365, SharePoint, Azure, Dynamics 365, Twitter, Dropbox, DocuSign, Gmail, MailChimp, WordPress, YouTube, and Zendesk. You can even connect to Azure Solutions, including Cognitive Services (Computer Vision, Text Analytics), QnA Maker, Microsoft Bot Framework, and Azure IoT Central, to seamlessly integrate these advanced capabilities into your apps.

**Custom connectors**

We offer a wide variety of connectors, but sometimes you might want to call APIs, services, and systems that aren’t available as prebuilt connectors. To support more tailored scenarios, you can build custom connectors with their own triggers and actions. These connectors are function-based—data is returned based on calling specific functions in the underlying service.

**Why build a connector?**

**Enterprise developers**

Custom connectors are a great solution for enterprise developers looking for connectivity to a service we don’t currently support or if they need connectivity to a custom or internal service.
Partners and ISVs
Partners and ISVs can leverage custom connectors to expand the number of services with which they integrate, provide extensibility to your customers needs, and to increase exposure and adoption.

As stated above, building a connector offers extensibility of your app through PowerApps, as well as automation and integration through Microsoft Flow and Logic Apps. The same connector can drive more usage of your service and your existing API without additional development.

Drive more usage
Increase reach, discoverability and usage of your service by publishing pre-defined task specific templates that integrate your app with our growing family of connectors.

The PowerApps and Microsoft Flow gallery of connectors and templates make it easy for your users to get started. Embedding of the Microsoft Flow experience within your app enables users to leverage pre-built templates from right within your application.

Expand the reach of your API
Enable power users to leverage your APIs and extend your solution without having to write code. Using simple clicks, a business user can create and share a multitude of solutions for organizational or personal use.

Custom connector architecture
Before a connector is built, we start with two components that need to be connected. On one side, we have the Power Platform and the Business Applications Platform; on the other side, we have our RESTful API. These RESTful APIs can be in the cloud or on premises. The custom connectors create a bridge between these two components. These connectors sit in a Microsoft-owned hosting environment.
Understanding connectors

Building blocks
There are three building blocks needed in order for Microsoft Flow or PowerApps to execute an operation:

- Connector ID (connector name)
- Operation to run
- Connection ID (reference to credentials that will be used to perform operation)

When creating a custom connector, the open API description of the RESTful API is transformed in such a manner that it describes this wrapper. As such, Microsoft Flow and PowerApps never actually talks directly to RESTful service.

Once these building blocks have been set up and the API call goes through, the hosting environment and the custom connector transform this information into a Host (Connector ID), Path (Operation), and Credentials (Connection ID), which are passed to the RESTful API. From there, the RESTful API responds back to the custom connector and the custom connector subsequently responds back to Microsoft Flow or PowerApps.

This additional level of abstraction provides greater flexibility, as we don’t need to require a standard way of describing APIs. Furthermore, it creates a more secure environment; your credentials never leave the client.

Using connectors
You use your custom connector like you use other public connectors. You create a connection to your API and call any operations that the API provides in the same way that you call operations for a Microsoft-managed connector.

Connectors you create in Microsoft Flow or PowerApps can be used in both services. Connectors you create in Logic Apps cannot be used directly in the other services. That said, you can easily recreate a connector by using the same OpenAPI definition or Postman collection you used to create the connector.

Using connectors within multiple products
PowerApps
PowerApps enables users to build cloud-connected and cross-platform business apps using clicks and minimal code. Create rich user experiences across the web, phones, and tablets. Assemble forms, add business logic, and take advantage of device capabilities with full creative freedom.

From the maker portal select the gear > Custom connectors > (+) Create custom connector.

Learn how to do more
Learn how to build Flows
Learn how to build manage your connections
Learn how to create an app from scratch
Learn how to use the formula builder
Learn how to use a custom connector from an app
Learn how to use a custom connector from a logic app
Understanding connectors

**Microsoft Flow**
Work smarter by building workflows and automating processes across your apps and services. Streamline notifications, sync data between systems, automate approvals, and more.

1. Create a new flow: My flows > (+) Create from blank.
2. Search all connectors and triggers > Add connector.
3. Select connector (Sign in to authorize based on security set).

**Logic Apps**
Logic Apps is the workflow engine for Microsoft Flow. It enables pro-developers to visually create or programmatically configure workflows in Azure. A connector in Logic Apps enables users to automate EAI, business to business (B2B) and business to consumer (B2C) scenarios while reaping the benefits of source control, testing, support, and operations.

In Logic Apps, you can use enterprise connectors to create logic app workflows and automate processes between cloud apps and cloud services.
Building a custom connector

**Step 1**
Build and secure your API

**Step 2**
Describe the API and define the custom connector
- Describe the API
- Building a custom connector

**Step 3**
Securing your custom connector
- Secure your Web API
- Secure your custom connector

**Step 4**
Testing your custom connector
- Test from connector settings
- Test with a new flow or PowerApp
**1 Build and secure your API**

A custom connector is a wrapper around a REST API that lets an underlying service talk to PowerApps, Microsoft Flow, and Logic Apps. So first, you need a fully-functioning API before you create a custom connector.

You can use any language and platform for your API, as long as it’s made available as a REST API. Here are a few examples:

- Publicly available APIs.
- An API that you create and deploy to any cloud hosting provider.
- A custom line-of-business API that's deployed to your network. You can connect to the API if it’s available over the public internet, or you can connect to it through a gateway (currently available in Microsoft Flow and PowerApps).

We recommend using one of these standard authentication methods for your APIs and connectors:

- Generic OAuth 2.0
- OAuth 2.0 for specific services, including Azure Active Directory (recommended)
- Basic authentication
- API Key

You can set up Azure Active Directory (AAD) authentication for your API in the Azure portal, so you don’t have to implement authentication through code. Alternatively, you can require and enforce authentication through your API's code.

**2 Describe the API and define the custom connector**

**Describe the API**

If you have an API with some type of authenticated access, the next thing to consider is how you describe your API's interface and operations so that PowerApps, Microsoft Flow, or Logic Apps can communicate with your API. After you decide on how to describe the API, you create the connector, which registers it with the appropriate service(s).
Building a custom connector

You can use one of the following approaches to describe your API:

- An OpenAPI definition (formerly known as a Swagger file).
- A Postman collection.
- In Microsoft Flow and PowerApps, you can also start from scratch in the custom connector wizard.

OpenAPI definitions and Postman collections use different formats, but both are language-agnostic, machine-readable documents that describe your API’s operations and parameters. You can generate these documents from various tools, based on the language and platform used by your API. Behind the scenes, PowerApps, Microsoft Flow, and Logic Apps use OpenAPI to define connectors.

Building a custom connector
The general process to build a connector involves several steps:

1. Open Microsoft Flow, go into the gear on the top right and select Custom Connectors.

2. Select (+) Create custom connector. From the dropdown, you can select the method from which you’d like to create your custom connector.

3. Import your custom connector (or Create from blank).

4. Once imported, you will be prompted to go through a series of steps to complete the setup: General > Security > Definition > Test.
Securing your custom connector

There are two primary concepts involved in securing your Web API:
1. Use OAuth for sensitive data.
2. Secure your Web API with an AAD (Azure Active Directory) Application (Identity) and a second AAD Application to secure custom connector.

This creates two disjointed entities that depend upon each other, keeping security definitions and entities separate. As such, the process for securing your custom connector involves creating two AAD applications: one to secure your Web API and a second to secure the custom connector. While you can use other authentication methods, we recommend AAD as it provides you the ability to have disjointed entities for both services.

Secure your Web API
1. Go to your web application and enable AAD authentication. Make sure you enable “Log in with Azure Active Directory.” Then configure, selecting the appropriate AAD app for your Web API.

Secure your custom connector
2. Create a new AAD application for your custom connector and give it delegated access to your first AAD app (which identifies your Web Service).
3. In Microsoft Flow, update your custom connector’s Security section. Under Authentication type, select OAuth 2.0, and as the Identity Provider, select Azure Active Directory.
4. Assign the AAD connector app to the custom connector using the following information.
   • Custom connector Client ID = AAD app’s Application ID.
   • Custom connector Client secret = AAD app > Key > New Key > Value.
   • Custom connector Resource URL = Application URL or Application ID of Web API AAD application.
Building a custom connector

5. Copy Redirect URL from the custom connector in Microsoft Flow to your custom connector AAD app > Settings > Reply URLs.

6. Enable Cross Origin Resource Sharing (CORS) on your website. This will help prevent man-in-the-middle attacks (Allowed origins: Enable everything or Specific environments).

7. In Microsoft Flow, update the connection. From the gear menu, select Connections, then select your connector. Click on the Auth key, and you will be prompted to sign in and to grant access. Then tokens will be stored in the connection.

4 Testing your custom connector

When testing your custom connector, you can test the connection from the custom connector settings or from testing the connector itself. These methods can be used to validate your API calls, as well as to do end-to-end testing with Microsoft Flow and PowerApps.

Test from connector settings
From you connector settings in Microsoft Flow, navigate to the Test tab, add a New connection, and select Test operation.

Test with a new flow or PowerApp
To test your connector, you can create a new flow. From the My flows page and select (+) Create from Blank. Choose a trigger and add an action. Search for your connector (which should show up alongside the public connectors). Once the flow is created, you can test the connector performance. Similarly, you can create a simple app in PowerApps, select the gear > Custom connectors > (+) Create custom connector. Once you've built your PowerApp, you can test the performance of your connector.
Publish a custom connector

Share your connector internally

Publish to the connector store
- Connector certification
- Certification criteria
- Submit your connector to Microsoft for certification

Publish to AppSource
- Create a managed solution package
- Validate your code
- Create a new offer
- Submit for publication
- Submit your solution
- Market and sell
Publish a custom connector

Once you’ve created your custom connector, it’s time to share it, and there are three ways you can do this depending on with whom you wish to share the connector and how it will be used.

1. The first is to share for internal use, either with an internal team or client.
2. The second option is to make it available through the connector gallery on the Microsoft Flow website. This process that involves certifying your connector with Microsoft.
3. The third option is to integrate the connector into an app and release that app through the AppSource.

Share your connector internally

You can share your connector with users in your organization in the same way that you share resources in PowerApps, Microsoft Flow, and Logic Apps. Although sharing is optional, you might have scenarios where you want to share your connectors with other users.

Publish to the connector store

Connector certification

As part of our third-party certification process, Microsoft will review the connector before publishing. This process validates the functionality of your connector and checks for technical and content compliance as well as scenario fit.

If your application is built on Azure, every user of Office 365 Enterprise Plans will get instant access to the connector for your app.

To make a custom connector publicly available for all users in PowerApps, Microsoft Flow, and Logic Apps, submit your connector to Microsoft for certification. Microsoft will review the connector and, if it meets certification criteria, will approve it for publishing.

Why should you certify a connector?

Certifying a connector and releasing it publicly offers extensibility of your app through PowerApps, as well as automation and integration through Microsoft Flow and Logic Apps. The same connector can drive more usage of your service and your existing API without additional development.
Publish a custom connector

Certification criteria
In order to qualify for certification, you must meet the following requirements:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Details</th>
<th>Required or Recommended</th>
</tr>
</thead>
<tbody>
<tr>
<td>Software-as-a-Service (SaaS) app</td>
<td>You must either own the underlying service or present explicit rights to use the API; and provide a user scenario that fits well with our products.</td>
<td>Required</td>
</tr>
<tr>
<td>Authentication type</td>
<td>Your API must support OAuth2, API Key, or Basic Authentication.</td>
<td>Required</td>
</tr>
<tr>
<td>Support</td>
<td>You must provide a support contact so that customers can get help.</td>
<td>Required</td>
</tr>
<tr>
<td>Availability and uptime</td>
<td>Your app has at least 99.9% uptime.</td>
<td>Recommended</td>
</tr>
</tbody>
</table>

Before Certification
Before submitting your connector for certification, review the functionality and content of your connector. It is required that you test your custom connector within PowerApps, Microsoft Flow, and Logic Apps to ensure the operations work as expected and the naming, descriptions etc. make sense to the end user.

Submit your connector to Microsoft for certification
Certification typically takes about two weeks from submitting the artifacts through publishing, depending on how well it meets the certification criteria. To apply for certification, follow these steps:

1. Nominate
   - Submit your nomination.
   - If your connector is approved, Microsoft will send the mutual Non-Disclosure Agreement and the Partner Agreement that are required to submit your connector for review.

2. Review
   Once the agreements are signed, send this information to your nomination contact for review:
   - The connector ID for your custom connector (the generated ID found in the URL when selecting your custom connector). Note this custom connector should have at least 10 successful calls per operation.
   - The OpenAPI specification file that describes your API.
   - The icon file (.png or .jpg) that represents your connector.
     - Your icon should have a ~160-pixel logo inside a 230-pixel square. A white logo on a colored background is preferred.
Publish a custom connector

• Your icon’s brand color in hexadecimal format, which should match the colored background in the icon file.

• A test account for validation.
  • If all operations require test parameters, provide test values for at least one operation.
  • If your connector uses OAuth, provide client ID/secrets for testing with https://*.consent.azure-apim.net/redirect as the whitelisted redirect URL. We’ll reach out for the production details.

• An email address for support or details on how customers can file support issues.

3. Publish

After we validate your connector’s functionality and content, we stage the connector for deployment across all products and regions.

By default, all connectors are published as “preview” and “premium.” A “premium” connector means users will need a paid subscription to leverage them. If the service your connector leverages is built on Azure, you can apply for listing your connector as a “standard” connector that’s available to all users with Office 365 Enterprise plans. For more details, ask your nomination contact.

4. Support/Updates

Once the connector is released publicly, you may monitor your connector and the related product forums to see if customers hit any issues or have feature requests.

Updates to an existing connector can be submitted through this certification process.

After the connector has been publicly available for some time, it can qualify to have the “preview” tag removed. For more details, ask your nomination contact.
Publish to AppSource

Custom connectors can be published through an integration app for a first-party Dynamics 365 app. The following details the requirements and steps to publish your app on AppSource.

<table>
<thead>
<tr>
<th>Schema</th>
<th>User interface</th>
<th>Analytics</th>
<th>Process/Code</th>
<th>Templates</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Entities</td>
<td>• Application ribbon</td>
<td>• Dashboards</td>
<td>• Processes</td>
<td>• Mail-merge</td>
<td>• Security roles</td>
</tr>
<tr>
<td>• Attributes</td>
<td>• SiteMap</td>
<td>• Reports</td>
<td>• Dialogs and workflows</td>
<td>• E-mail</td>
<td>• Field level security profiles</td>
</tr>
<tr>
<td>• Relationships</td>
<td>• Forms</td>
<td>• Visualizations</td>
<td>• Plug-ins</td>
<td>• Contract</td>
<td></td>
</tr>
<tr>
<td>• Global option sets</td>
<td>• Entity ribbons</td>
<td></td>
<td>• Assemblies</td>
<td>• Article</td>
<td></td>
</tr>
<tr>
<td>• Web resources</td>
<td></td>
<td></td>
<td>• Processing steps</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 Create a managed solution package

There are two types of solutions for Dynamics 365 Customer Engagement: managed and unmanaged. An unmanaged solution is one that is still under development or isn’t intended to be distributed; unmanaged solutions can still be edited. Once your unmanaged solution is ready to be distributed, you must export the unmanaged solution as a managed solution. A managed solution is a completed solution—with publishable code—that is intended to be distributed and installed by users.

Export solution to Package Deployer

When you are ready to publish to AppSource, you must create an AppSource Package. Export your unmanaged solution as a managed solution, including customizations and metadata, to the Package Deployer. Dynamics 365 provides you with a Visual Studio template for exporting packages that can be used with the Package Deployer tool.

Packaging for PowerApps and Dynamics 365.

Both canvas and model-driven apps need to be bundled as a managed solution in order to be published on AppSource.
Create a package using Package Deployer

Using Package Deployer, create a package (.zip) with your assets. The Package Deployer creates a solution package with the structure needed to submit to Microsoft for certification.

Creating a package involves creating a project using the template, adding your files to the project, updating the HTML files, specifying the configuration values for the package, and defining custom code for your package.

An AppSource package consists of:

- **Package file**: A package file used by Package Deployer to deploy your solutions and demo configuration data into multiple languages.
- **[Content_TYPES].xml**: File that provides MIME type information of the file type extensions included in the AppSource package. Typically, these are .config, .dll, .exe, .xml, and .zip file types, but you can add almost any file type that is supported by Windows.
- **Icon file**: An image file for the AppSource package icon; size should be 32x32 pixels. Valid image formats are PNG and JPG.
- **HTML file**: File containing your License terms.
- **Input.xml**: File that describes the assets in your AppSource package.

Upload package to Azure

Once you have created your solution package, you must upload it to Azure. Before you upload, you should download and install the Microsoft Azure Storage Explorer, which enables you to manage the contents of your Azure storage account easily.
Validate your code

QA your code
Before submitting your code for certification, it is always recommended to thoroughly QA your code, especially in the following areas:

- **Usage**: Improper usage of a particular API, pattern, or configuration.
- **Design**: Design flaws in a customization.
- **Performance**: Customization or pattern that may produce a negative effect on performance in areas such as memory management, CPU utilization, network traffic, or user experience.
- **Security**: Potential vulnerabilities in a customization that could be exploited in a runtime environment.
- **Upgrade Readiness**: Customization or pattern that may increase the risk of having an unsuccessful version upgrade.
- **Online Migration**: Customization or pattern that may increase the risk of having an unsuccessful online migration.
- **Maintainability**: Customization that unnecessarily increases the amount of developer effort required to make changes, the frequency of required changes, or the chance of introducing regressions.
- **Supportability**: Customization or pattern that falls outside the boundaries of published supportability statements, including usage of removed APIs or implementation of forbidden techniques.

Additionally, we recommend that you follow these best practices when building your app, as it can expedite the certification process.

Validate your PowerApp
Leverage the Solution checker feature to perform a rich static analysis check on your solutions against a set of best practice rules and quickly identify these problematic patterns. After the check completes, you receive a detailed report that lists the identified issues, the affected components and code, and links to documentation that describes how to resolve each issue.

1. In PowerApps, select the **Common Data Service** where you want to enable the Solution checker.
2. On the left navigation pane, select **Solutions**.
3. On the toolbar, select **Solution checker** and then **Install**.
4. Once installed, navigate back to the **Solutions** menu, open the dropdown menu (...), and select **Solution checker > Run**.
5. **View the report** when the check is complete.
Create a new offer

Create a new offer in the Partner Center
Once you have resolved any issues identified by the Solution checker, you can move on to creating an offer in the Partner Center (PC).

Each app in AppSource corresponds to an Offer in PC. To submit a new app to be published on AppSource, you must create a New offer in PC. There are different offer types based on different types of apps. When creating an offer, you will be asked to provide information, including:

- **Offer setup**: Setup details, Lead Management, ISV Program.
- **Properties**: Category, industries, version, T&C, market only change.
- **Offer listing**: Name, description, contacts, documents, logos, screenshots, videos, etc.
- **Availability**: Markets, key.
- **Technical configuration**: License model, package URL.
- **Supplemental content**: Functional specification document.

About Test Drive
AppSource supports PowerApps Test Drive solutions as a way for you to share apps with customers and generate leads for your business.

TEST DRIVE ➤
Publish a custom connector

4 Submit for publication

Click Publish to submit your app for approval
Once you have completed filling in the offer details, click Publish to begin the publishing approval process. Progress can be tracked on the offer page.

Steps to publish
1. Automated validation: This step includes Test Drive data validation (<5 min), Test Drive provisioning (<30 min), and Lead management validation and registration (<15 min).
2. Certification: This step leads to manual validation (<2 business days).
3. Preview creation: Preview creation (<1 hour).
4. Publisher sign-off
5. Publish: This step consists of Test Drive data validation (<5 min), Test Drive provisioning (<30 min), Lead management validation and registration (<15 min), and Final publish (<30 min).

5 Publish your solution

View and publish your solution
Once your app has been approved for publication on AppSource by Microsoft, you will receive a preview link to your offer, where you will be able to view your offer on AppSource and test as if it were live. Once you are ready, you can click on Go live, at which time it will become publicly available on AppSource.

Certify/Recertify your application
To complete publishing your app to AppSource, it needs to go through the certification process. Click Save and Publish.

You need to recertify your apps every 6 months to keep them live on AppSource. Follow the steps to certify/recertify your application to AppSource.
Here are a few criteria to get your apps recertified:
- Check and verify if your app is on the latest Dynamics 365 version.
- Check if the Dynamics 365 environment is utilizing Unified Interface.

6 Monitor performance

Monitor performance and collect leads
Within Partner Center (PC), under the Commercial Marketplace Analyze tab, you can view app performance. You will also receive leads from users who registered for Test Drive or Trial, requested to be contacted through the Contact me form, or opted to share their information by selecting Get it now.

ISV Studio
ISV Studio is designed to become the go-to Power Platform destination for Independent Software Vendors (ISVs) to monitor and manage their applications. ISV Studio provides a consolidated cross-tenant view of all the applications ISVs are publishing on AppSource. It helps the ISVs monitor and support their published apps with the help of insights into installation error messages, number of install attempts (Success vs Failures) by tenant name, tenant and instance locations, prod vs sandbox installations, package versions by tenants, etc.

7 Market and sell

Leverage Microsoft support to market and sell your app
As soon as your app is published, you can take advantage of Microsoft’s Go-To-Market Services, which will help you promote and sell your app. You may also be eligible to participate in other partner programs, such as the IP Co-Sell program.
ISV Cloud Embed Program

If you have built or intend to build a vertical/industry-focused solution on top of Dynamics 365, or a horizontal solution on PowerApps, and publish on AppSource, the Microsoft ISV Cloud Embed program simplifies app development time, lowers costs, and helps your business grow. The program allows ISV partners to focus on continuous innovation and rapidly build business applications by extending Dynamics 365 applications or building on PowerApps; benefit from a growing community of Office 365 and Dynamics 365 customers through AppSource; receive critical go-to-market support; and delivers increased benefits culminating in co-sell support from one of the world’s largest enterprise salesforces.

The requirements for the program are:
1. End-to-end solution built on Dynamics 365 or Power Platform (PowerApps).
2. Solution published on AppSource.
3. Become CSP Direct Bill partner.

It is important to note that you can only sell the Embedded SKUs with your IP as a Unified Solution; you cannot sell the Embedded SKUs separate from your IP, and you will have to give one price point to the entire Unified Solution.

Getting started
Any ISV that builds qualified, finished applications can participate in the ISV Cloud Embed Program at different levels with different benefits based on the partner’s preference. To learn more about the program, please review the ISV Cloud Embed Handbook. To become a part of this program and guarantee a rich customer experience, please sign up through this simple online form.
Additional resources
Additional resources

Learn more about connectors: Documentation
Build and certify your Flow, PowerApps & Logic Apps Connector: Article
Connectors FAQ: FAQ

Best practices and common use cases
Community content: Community apps gallery | Working with data | Design discussion
Real world solutions using PowerApps: Video
UX/UI design tips: Video | UX patterns | Managing fonts and colors
Customize list forms in SharePoint with PowerApps: Blog post | Demo | Documentation
Access web apps migration to PowerApps: Whitepaper
Approval workflows: Blog post | Documentation | Guided learning
Using the on-premises gateway: Overview | Installation and FAQ | Proxy configuration
Creating dialogs in PowerApps: Blog post | Video
Display a map using the Image control: Step-by-step walkthrough | Video
Send an email from your PowerApps app: Video
Connect to Microsoft Cognitive Services: Video
Implement role-based security: Blog post
Build a customized interactive calendar: Step-by-step guide
Notify user that new data is available: Step-by-step guide | Push notifications feature
Deep link into an app using URL parameters: Step-by-step guide
Audit scenario solution: Blog post from community member
Performance considerations when working with PowerApps: Blog post

Learning resources
PowerApps latest feature updates: Blog post | Release notes
Microsoft Flow latest feature updates: Blog post | Release notes
Browse presentations from the Ignite 2018 conference: Blog post with curated links to relevant sessions
Browse Microsoft Business Applications Summit 2018 conference: Blog post with curated list to relevant sessions
Browse How-to videos: PowerApps video gallery | Microsoft Flow Webinars and Videos gallery
Get expert help from partners: Partners
Take up additional Labs Power Platform Labs and Challenges
SharePoint welcomes PowerApps and Flow
Customizing SharePoint forms using PowerApps: Ignite presentation
Introduction to Microsoft PowerApps for Access web apps developers

Stay connected
Ways to collaborate with the PowerApps team: Blog post
Stay updated: Product team blog | Webinars
Community sites: Forums | Community blog | Best practices and app gallery
Get support: Blog post | Support page | Submit an idea

Other tools
Microsoft App in a Day Lab
PowerApps and Microsoft Flow Learning Resources
Additional resources

Getting Started Guides
Check out our other Getting Started Guides.
Disclaimer

© 2019 Microsoft. All rights reserved.

This document is provided “as-is.” Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and are fictitious. No real association is intended or inferred.

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.