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 Cloud Partner Portal (CPP). If you are already a Microsoft partner with Cloud Partner Portal (CPP) access, you can skip this section.

**Understanding the Power Platform**
The second section takes a look at the Power Platform to provide context and clarity around the structure and available tools.

**Understanding connectors**
The third section provides an overview of connectors to help you understand what connectors are and how they work.

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

**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 applica-
tions—including Finance and Operations, CRM, Office 365, and LinkedIn—the
true power of the platform is in how it all works together through our Com-
mon Data Service.

The Common Data Service (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 consisten-
cy 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 ty-
ing 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: Extend, Build, and Connect.

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 Opera-
tions. Extend scenarios include creating industry or vertical customizations
to our first-party apps.

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

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

Build
PowerApps Plan 2 is the minimum
required plan. Refer PowerApps pric-
ing for more details.

Connect
PowerApps Plan 1 is the minimum
required plan. Refer PowerApps pric-
ing for more details.
Contents

4

Prerequisites
Become a Microsoft partner
Register as a publisher
Create a seller account
Sign up for Power Platform

8

Understanding the Power Platform
Parts of the Power Platform
Common Data Service
Model-driven apps vs. Canvas apps
Ways to build PowerApps
Programming models

8

Understanding connectors
Understanding connectors
Custom connector architecture
Using connectors

14

Building a custom connector
Steps to build a custom connector
Securing your custom connector

14

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

29

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 the Cloud Partner Portal
- Sign up for the Dynamics Insider Program

Step 3
Create a seller account
- Create a Microsoft Developer account

If you are already a Microsoft Partner with Cloud Partner Portal 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.

Create a Microsoft work account
Before you begin, you must create a Microsoft work account. The same account should be used as you sign up for subsequent programs to ensure all of your privileges are centralized under a single account ID. You can register your email 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.
Prerequisites

Set up your Partner Center account
Once you have joined the Microsoft Partner Network (MPN), you can set up your Partner Center account. Your Partner Center account provides you with access to pricing information, tools and services, and enables you to manage admin credentials for your company's work account. Partner Center 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 the Cloud Partner Portal
The first step to becoming a publisher is to register in the Cloud Partner Portal (CPP). The Cloud Partner Portal (CPP) is where you submit your apps for publication, publish and promote your apps, and manage your offers.

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

Once registered, you can access the Cloud Partner Portal (CPP).

Starting in H2 2019, the CPP process will be moved to Partner Center.

In the future, you will be able to select PowerApps. Currently, you must select Customer Engagement.

If you’re already registered in the Cloud Partner Portal, you don’t need to register again; however, we recommend you submit the form to connect with a member of our team.

Learn how to manage your Partner Center account.

DOCUMENTATION
Prerequisites

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.

Create a seller account
Creating a seller account gives you the necessary resources to market and sell your solutions on AppSource.

Create a Microsoft Developer account
To market and sell your solutions on the AppSource marketplace, you must create a Microsoft Developer account in the Microsoft Developer Center. When registering, be sure to use the same Microsoft account ID that was used for the Cloud Partner Portal to ensure that your Microsoft Developer and Cloud Partner Portal accounts are appropriately linked.
Understanding the Power Platform

Parts of the Power Platform
• PowerApps
• Microsoft Flow
• Power BI

Common Data Service
Understanding the Power Platform

Parts of the Power Platform
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. web.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.
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:
Understanding the Power Platform

• **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 build in Flow, these workflows and connections can be leveraged the extend the functionality of your PowerApps apps. 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 apps, 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. Common Data Service for Apps lets you securely store and manage data used by business applications. Data within the Common Data Service 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.

The Common Data Service 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 Common Data Service for Apps to store and secure data used by the applications. This means you can build apps using PowerApps and Common Data Service 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 Common Data Service for Apps.
Understanding connectors

Step 1  Understanding connectors
• What is a connector?
• What can connectors do?
• Components of a connector
• Public connectors vs. Custom connectors
• Why build a connector?

Step 2  Custom connector architecture

Step 3  Using connectors
• Connectors are available for use within multiple products
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 in them.

What can a connector do?
Developing a single connector enables an integration for three different Microsoft products – Flow, PowerApps, and Logic Apps.

Flow
Using 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
Understanding connectors

**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.

**Logic Apps**

Logic Apps is the workflow engine that powers 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**
  - **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 Swagger.
Understanding connectors

Public connectors vs. Custom connectors

Public connectors
Public connectors are available to all users. With a growing collection of over 200 connectors, 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, such as Cognitive Services (Computer Vision, Text Analytics), QnA Maker, Microsoft Bot Framework, and Azure IoT Central to seamlessly integrated 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 curently 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 they integrate with, 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 and automation and integration through 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 Flow and PowerApps gallery of connectors and templates make it easy for your users to get started. Embedding of the Flow experience within your app enables users to leverage pre-built templates from right within your application.
Understanding connectors

**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 like the one shown below, for organizational or personal use.

2 **Custom connector architecture**
There are three building blocks: Connector ID + Operation + Connection ID

When creating Custom connector > Take open API description off of RESTful API > transforms it such that it describes the wrapper. Flow and PowerAPps never talks directly to RESTful service

When call happens, host environment and custom connector transforms information to Host (Connector ID) + Path (Operation) + Credentials (Connection ID) > RESTful API responds to custom connector > Responds to Flow/PowerAPps

3 **Using connectors**
You use your custom connector like you use Microsoft-managed 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 Microsoft-managed connectors.

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.
Understanding connectors

Using connectors within multiple products

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

1) Create new flow [My flows > Create from blank] > Search all connectors and triggers
2) Select connector (Sign in to authorize based on security set)

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.

-In maker portal > See custom connectors

Logic Apps
Logic Apps is the workflow engine for Flow. It enables pro-developers to visually create or programatically 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.

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
Building a custom connector

Step 1
Building a custom connector

Step 2
Secure your connector

Step 3
Testing your connector
Building a custom connector

1 Building a custom connector

Building a Custom Connector
The first step to building an API Connector is to build a fully functional Custom Connector within the Flow or PowerApps experience. An API Connector is nothing more than a Custom Connector that is visible to all users of PowerApps and Flow.

The general process to build a connector involves multiple steps:

Develop
1. Build, secure, and describe your API.
2. Import from existing artifacts (Open API, Postman) or build from scratch.

Test
1. Validate your API calls.
2. Use in Flow and PowerApps for E2E testing.

Share
1. Allow other users within your organization to use your connector.

Certify (if public):
1. Share your connector with all users of Flow and PowerApps.

For software that operates outside the application:
- Use traditional methods to package and install your application, such as an installer program.

If your application consists only of Dynamics 365 solution components:
- Import directly into Dynamics 365. You won’t have to create an installer program.

If your extensions include a combination of Dynamics 365 solution components and external components
- You’ll need an installer.

Sometimes though, 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. The following diagram shows a custom connector for an API that detects sentiment in text. We’ll learn more about this scenario shortly.
Building a custom connector

The general process to build a connector involves multiple steps:

1) Open Flow > Go to gear > Custom Connectors

2) Create custom connector

3) [1. General] Add collection (Postman)
   - Option (Connect via on-premises data gateway

4) [2. Security] Set Authentication type

5) [3. Definition] Create definitions > Create connector

6) [4. Test]

The first step to building an API Connector is to build a fully functional Custom Connector within the Flow or PowerApps experience. An API Connector is nothing more than a Custom Connector that is visible to all users of PowerApps and Flow.
Securing your custom connector

Two concepts:
1. Use Oauth for sensitive data
2. Secure Web API with AAD (Azure Active Directory) Application (Identity) and a second AAD Application to custom connector

-Two disjointed entities that depend upon each other

Steps:
1. Go to web application and enabled AAD authentication
2. Create new AAD application for connector (give it delegated access to ADD app for web service)
3. Update connector > Security > Choose Oauth 2.0 > Azure Active Directory
4. Assign ADD connector app to Connector app
5. CC Client ID = ADD Application ID
6. (ADD Create Key>) Value = CC Client secret
7. CC Resource URL = Application URL or Application ID of Web API ADD application
8. Copy Redirect URL from CC = Copy to ADD CC app>Settings>Reply URL
9. Enable Cross Origin Resource Sharing (CORS) (Enable everything or specific environments)
10. Update connection (Settings > Auth) > Prompts access > Tokens then stored in the connection
Publish a custom connector

Share your connector internally

Publish to the connector store

Publish to AppSource
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 store on the Microsoft Flow website. This is a process that involves certifying your connector.
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 Logic Apps, Microsoft Flow, or PowerApps. Although sharing is optional, you might have scenarios where you want to share your connectors with other users.

Publish to the connector store

Submit for Certification

As part of our 3rd 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 Azure Logic Apps, Microsoft Flow, and Microsoft PowerApps, submit your connector to Microsoft for certification. Microsoft will review the connector and, if it meets certification criteria, will approve it for publishing. See the full list of publicly available connectors.

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 Azure Logic Apps and Microsoft Flow. 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 Flow, PowerApps 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” connectors 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 step 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>
</tbody>
</table>
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.

Storage account - blob, file, table, queue
Quickstart tutorial
2 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 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 issues identified, the components and code affected, 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 an offer

Create a new offer in the Cloud Partner Portal
Once you resolved any issues identified by the Solution checker, you can move on to creating an offer in the Cloud Partner Portal.

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

- **Offer Settings**: Offer ID, Publisher ID, Name.
- **Technical Info**: Metadata (e.g., categories, regions, marketing artifacts, and Azure storage URL of the solution package).
- **Test Drive**: Provide a test environment where users can use and explore a functional version of the app that has been preconfigured with demo data.
- **Storefront Details**: Offer summary, Offer description, Industries, Categories, Terms, etc.
- **Contacts**: Engineering Contact, Support Contact.

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.
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 publication**
1. **Validate prerequisites**: Offer settings provided are validated. (<15 min)
2. **Test Drive Validate**: Microsoft validates the Test Drive can be deployed and be replicated. (<2 hours)
3. **Certify Package**: This process will encompass certifying your package for deployment. (~15 days) Precertification process can expedite this step.
4. **Provision Package**: When complete, we will have deployed your package for use in the Regions requested. (~4 min)
5. **Lead management validation and registration**: Microsoft validates and registers lead management details. (<15 min)
6. **AppSource Packaging**: Offer is packaged to show up on AppSource. (<1 hour)
7. **Publisher signoff**: Offer is available to preview. Ensure that everything is correct before making your offer live.

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 were live. Once you are ready, you can set it to go live, at which time it will become publicly available on AppSource.
Monitor performance and collect leads
Within the offer page in the Cloud Partner Portal, 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.”

6 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.

Learn more about our Go-To-Market Services.
ISV Cloud Embed Program

If you have built or intend to build a vertical/industry focused solution on top of Dynamics 365 (or 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 ISV Cloud Embed at different levels and benefits based on the partner’s preference. To learn more about the program, please review the program Overview, Licensing guide, and 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
Getting Started Guides
Interest in building apps on our Business Application Platform? 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.