Microsoft
Power Apps

Streamline business processes with Microsoft Teams and Power Apps

VERSION: 1.0

AUTHOR: MATTHEW BOLAÑOS
COMPANY: Microsoft

RELEASED: November 2019

Copyright

Contents

Introduction

Power Apps is a high-productivity application development platform from Microsoft. The platform can be used to customize everything from simple SharePoint forms to immersive end-to-end solutions. Combined with Microsoft Teams, Power Apps can be used to build a modern workplace through custom tabs and apps in the app bar all with little to no code.

Purpose of this white paper

This white paper is targeted towards administrators and customizers responsible for planning, creating, deploying, and supporting applications built for Microsoft Teams on the Power Apps platform. The goal of the paper is to help you understand how to approach customizing Microsoft Teams, how to add Power Apps to Microsoft Teams, and how to administer the apps once they have been added.

Where possible, we will help you develop best practices for your organization to ensure successful deployments and high productivity for users using the platform.

Scope of this whitepaper

Unless specifically noted, all features mentioned in this white paper are available as of November 2019.

The following topics are out of scope for this white paper:

- Power BI, Power Automate, and other parts of the broader Microsoft Power platform.
- Power Apps fundamentals for building applications.
- ISV deployment scenarios, which are handled differently from enterprise deployment scenarios.
- Performance tuning of applications.
- Full deployment and management of first party Dynamics 365 applications.
- Third party solutions which integrate with Power Apps.
- Development of other types of apps using the Microsoft Teams developer platform.

Overview of the modern workplace

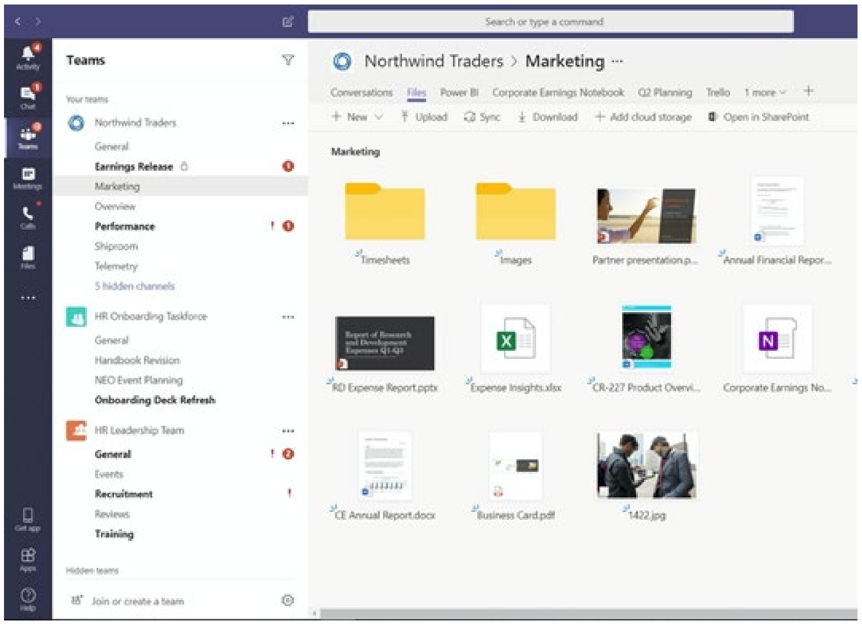
While using legacy systems, users previously had to jump between several different systems through a series of bookmarks or company portals. This constant switching can be incredibly disruptive. According to a June 2015 UC Irvine study, “The Cost of Interrupted Work: More Speed and Stress,” it can take upwards of 23 minutes for a person to regain productivity after such an interruption.

Additionally, many employees report having to complete routine tasks that limit the time they can spend on more creative and innovative tasks. A McKinsey Global Institute report claimed that employees spend 61% of their time on these routine tasks that could potentially be automated.

Both the productivity hit from constant context switching and the time constraints caused by routine tasks can quickly add up. A modern workplace aims to address these deficiencies in the status quo by providing a seamless experience that avoids disrupting employees, while simultaneously automating redundant or routine tasks that get in the way of much more valuable work.

Microsoft Teams is the hub for teamwork

Microsoft believes Microsoft Teams is the home for the modern workplace of the future. Out of the box, Microsoft Teams brings together the most common tasks that employees need under a single roof, such as chats, meetings, calls, and the productivity suite of Office 365. By combining these together into a sole product, employees can avoid having to constantly switch between various contexts. Instead, they can spend their time within a single team or channel that effortlessly brings together all the relevant information in-context.



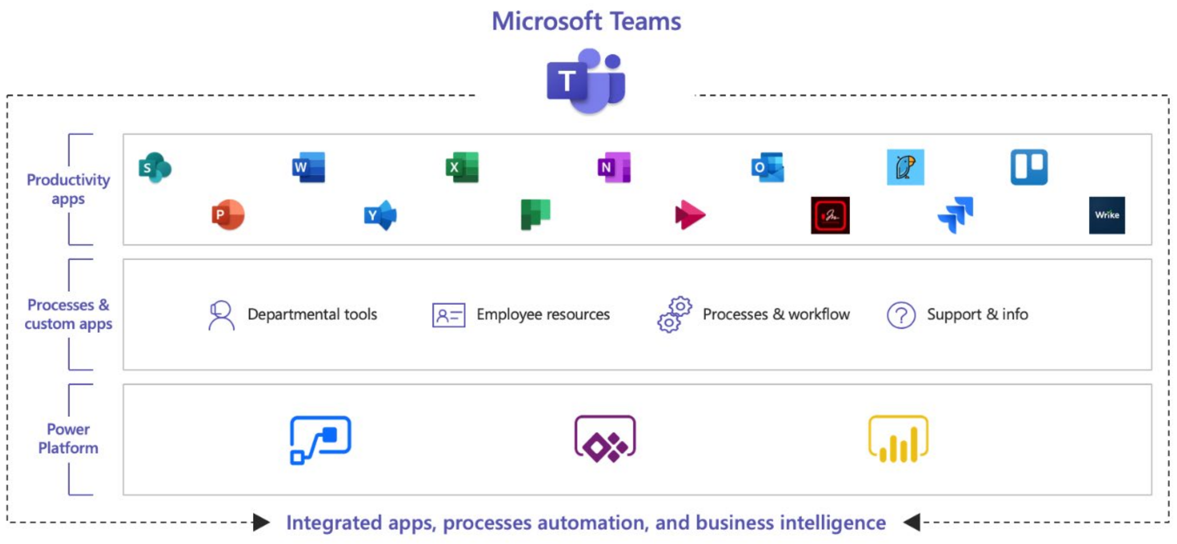
Microsoft Teams groups all the information you need for a particular context within various tabs in a channel.

The Microsoft Power Platform can augment this hub

Not all tasks, however, can come fully formed out-of-the-box. There will always be business processes that are unique to a company that require tailored solutions. This is where the Power Platform can come in to fill those gaps.

With tools like Power Apps and Power Automate, companies can build custom applications that can automate routine processes and supply a structure where there previously was none. And with Microsoft Teams, employees can use these custom apps all while taking advantage of the conversational nature of Microsoft Teams.

This is a radical departure from the past, where productivity software like email lived separately from business applications. The days of constantly switching between various tools is over now that companies can integrate *all* these tools into a single channel in Microsoft Teams.



Microsoft Teams can combine capabilities from a wide spectrum of tools to create a fully integrated experience.

Usage scenario

To better understand how you can leverage Power Apps within Microsoft Teams in your own company, we have supplied an example scenario that describes how you can unify disjointed experiences within a single hub in Microsoft Teams.

Collaborating as a team

As a hub for teamwork, the most common scenarios of Microsoft Teams are to bring together all the necessary tools a team would use within a single hub. The following scenarios show how groups across sales, engineering, and education can use Microsoft Teams and Power Apps together.

Completing a sales cycle

A sales team typically uses several different tools to complete a single sales cycle. These include tools to facilitate their internal conversations, the various PowerPoint decks they present to the customer, a customer relationship management (CRM) product, along with various reporting tools and approval processes.

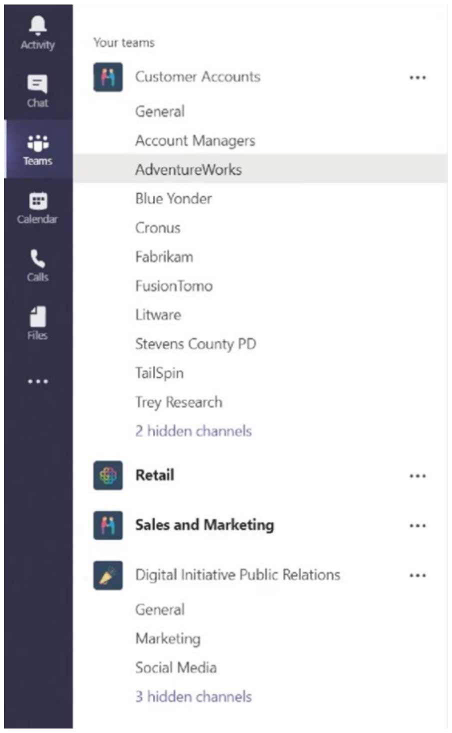
To provide structure to this scenario, we will follow a typical sales cycle:

1. Drafting the sales bid
2. Creating a sales opportunity
3. Drafting the customer order
4. Analyzing existing data about the customer
5. Submitting the proposal for approval
6. Meet with the customer to close the deal

Drafting a sales bid

Most projects start informally as a conversation between two or more colleagues. During this time, Team members can use the chat functionality within Microsoft Teams to discuss specifics about the upcoming deal and what strategies they want to use to close it.

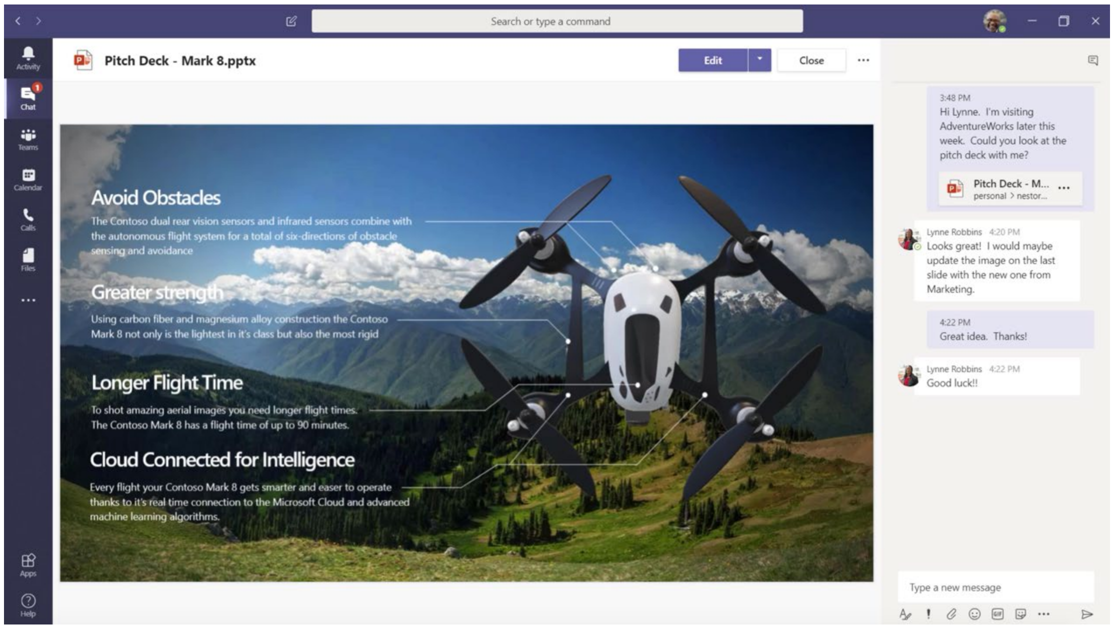
To organize their conversations and to supply a home for future work, they can create a central team for all their customer accounts where they can then enumerate all their accounts as individual channels. These channels make it easy to find relevant information in the future and helps contextualizes their work.



Each channel under the Customer Accounts team is a different account

Not long after chatting with the others on the team, they can start pulling together the collateral that they will use while meeting with the customer.

For example, they can open a PowerPoint deck directly within Microsoft Teams and start conversing about it together as they review it as a team. If any changes are necessary, they can even edit the deck inline within Microsoft Teams.


Pitch Deck - Mark 8.pptx

Avoid Obstacles

The Contoso dual rear vision sensors and infrared sensors combine with the autonomous flight system for a total of six-directions of obstacle sensing and avoidance

Greater strength

Using carbon fiber and magnesium alloy construction the Contoso Mark 8 not only is the lightest in it's class but also the most rigid

Longer Flight Time

To shot amazing aerial images you need longer flight times. The Contoso Mark 8 has a flight time of up to 90 minutes.

Cloud Connected for Intelligence

Every flight your Contoso Mark 8 gets smarter and easer to operate thanks to it's real time connection to the Microsoft Cloud and advanced machine learning algorithms.

3:48 PM
Hi Lynne. I'm visiting AdventureWorks later this week. Could you look at the pitch deck with me?

Lynne Robbins 4:20 PM
Looks great! I would maybe update the image on the last slide with the new one from Marketing.

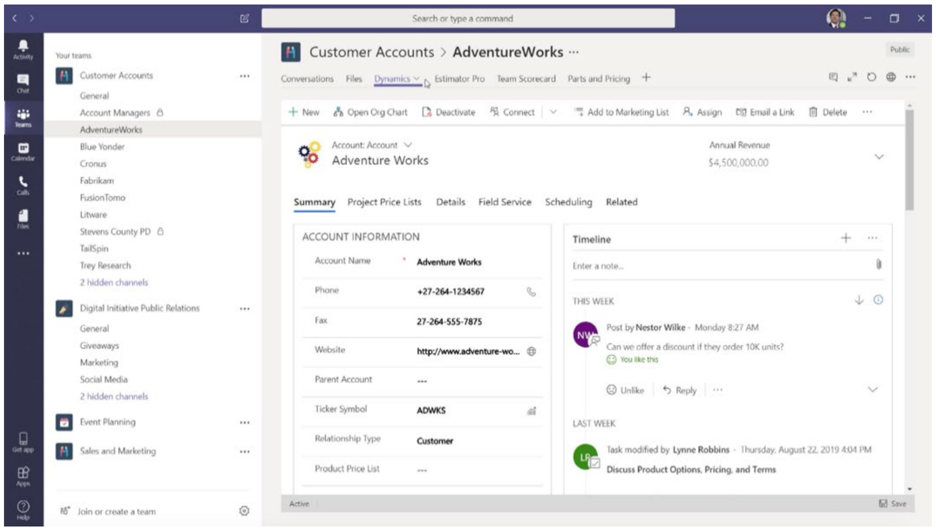
4:22 PM
Great idea. Thanks!

Lynne Robbins 4:22 PM
Good luck!!

Two colleagues discuss necessary changes to a deck before sharing it with a customer.

Creating a sales opportunity

Before getting too deep into the deal, the sales team should input all the relevant information into their CRM system. They are using Dynamics 365 for Sales, so it is extremely easy to embed a view of their current customer into Microsoft Teams.


Customer Accounts > AdventureWorks

Account: Account
Adventure Works

Annual Revenue
$4,500,000.00

ACCOUNT INFORMATION

Account Name	Adventure Works
Phone	+27-264-1234567
Fax	27-264-555-7875
Website	http://www.adventure-wo...
Parent Account	---
Ticker Symbol	ADWKS
Relationship Type	Customer
Product Price List	---

Timeline

THIS WEEK

Post by Nestor Wilke - Monday 8:27 AM
Can we offer a discount if they order 10K units?
You like this

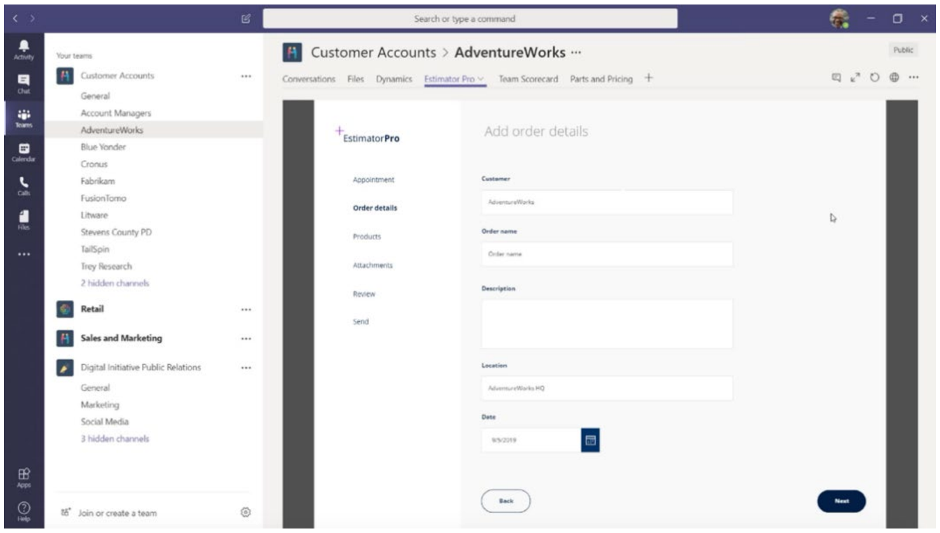
LAST WEEK

Task modified by Lynne Robbins - Thursday, August 22, 2019 4:04 PM
Discuss Product Options, Pricing, and Terms

A team reviews the Adventure Works account in Dynamics 365 for Sales.

Drafting the customer order

Once the team irons out the specifics of the deal, they can start creating an estimate for the order. Every company has a different order or estimation process, so this is the first place where the Power Platform can come in to help. For example, a company could create a canvas app that can collect all the relevant information before calculating the cost and saving the data.



Add order details

Customer: AdventureWorks

Order name

Description

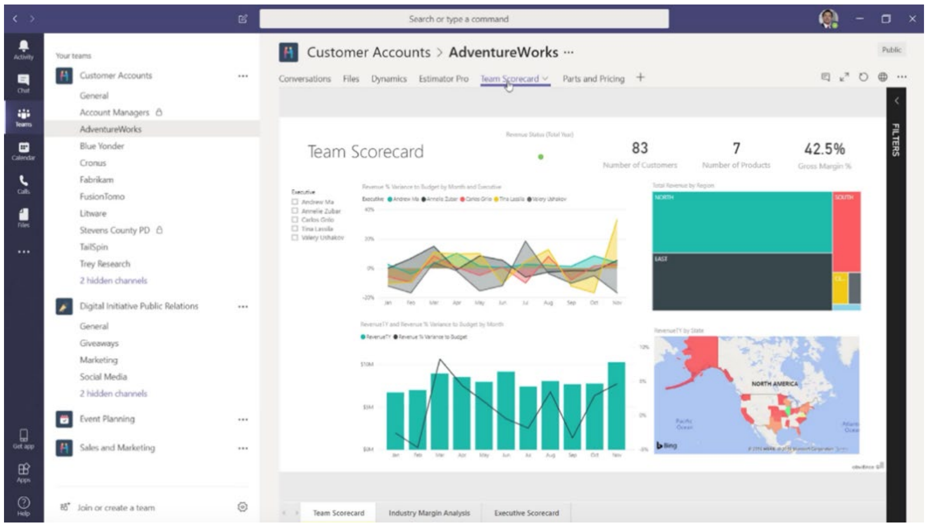
Location: AdventureWorks HQ

Date: 9/5/2019

A team uses a Power Apps form within the Adventure Works channel to create an estimate.

Analysing existing data about the customer

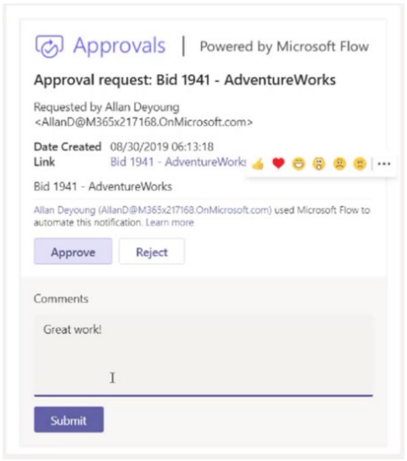
To better understand the customer before meeting face-to-face, it would also be beneficial to see existing data about the customer. How many orders have they placed in the past? How loyal are they? Are they likely to churn? For these needs, the team can use Power BI to view historical data about the customer.



A team uses Power BI to see the current health of the customer relationship.

Submitting the proposal for approval

Now that the team has armed themselves with an estimate, prior data, and a solid deck, they are now ready to get final approval of their sales bid from their manager. For this last step, they can use Power Automate automated approvals. These approvals can come directly into Microsoft Teams so that the manager never has to leave.



Approval request: Bid 1941 - AdventureWorks

Requested by Allan Deyoung
<AllanD@M365x217168.OnMicrosoft.com>

Date Created 08/30/2019 06:13:18
Link Bid 1941 - AdventureWorks

Bid 1941 - AdventureWorks

Allan Deyoung (AllanD@M365x217168.OnMicrosoft.com) used Microsoft Flow to automate this notification. Learn more

Great work!

A sales manager approves the sales bid.

Meet with the customer to close the deal

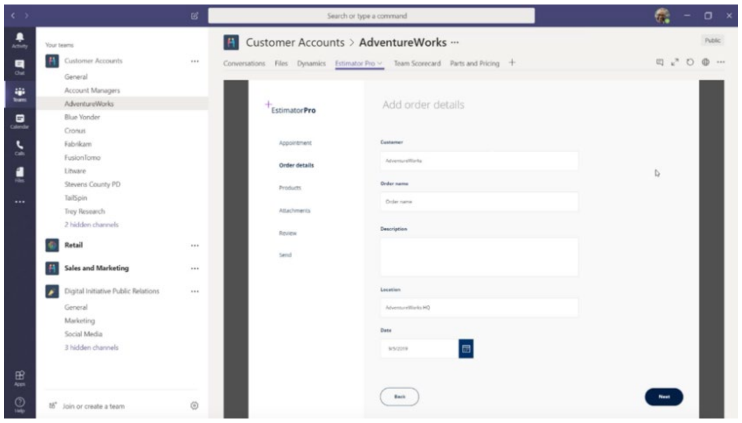
Finally, the sales team can use the meeting and call features baked within Microsoft Teams to reach out to the customer to close the deal.

Types of apps

Within Microsoft Teams and Power Apps, there are several types of apps. This section reviews them all so that you can better understand how the related and work with each other.

The different app modes in Microsoft Teams

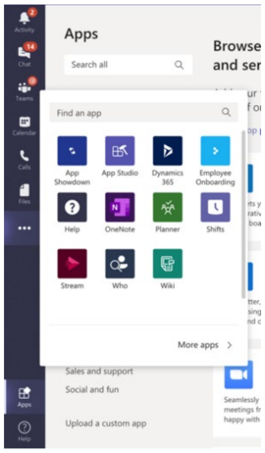
There are fundamentally two different app modes within Microsoft Teams: tab apps and personal apps. Tab apps are by far the most popular use of apps within Microsoft Teams. They can be pinned at the top of a channel to provide quick access to a collaborative workspace for a team.



Example of a tab app within a channel.

Personal apps, on the other hand, are applications that are specific to a single person. Typically, these are apps that the IT admin thinks that everyone in the company should be using. Instead of living within a channel or a conversation, they live in the app bar. Examples include time tracking, field service applications, and HR related apps. As of October 2019, only canvas apps are currently supported as personal apps.

By default, new personal apps will appear in the overflow region, but you can also pin applications to the app bar.

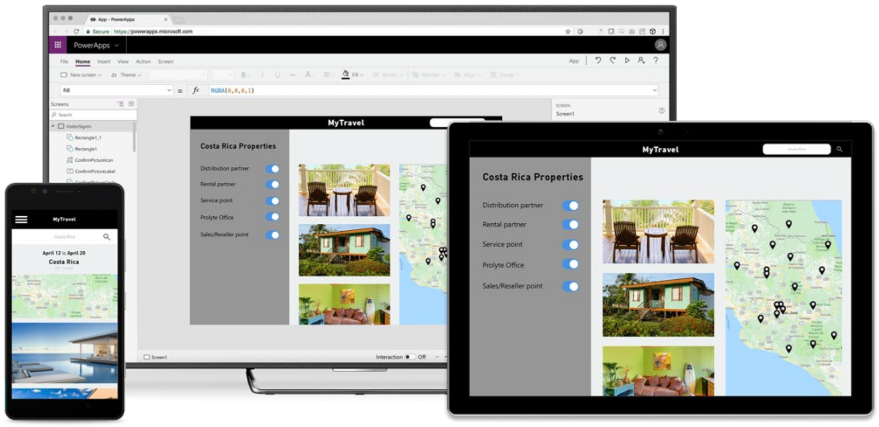


Example of personal apps in the app bar.

The different types of apps in Power Apps

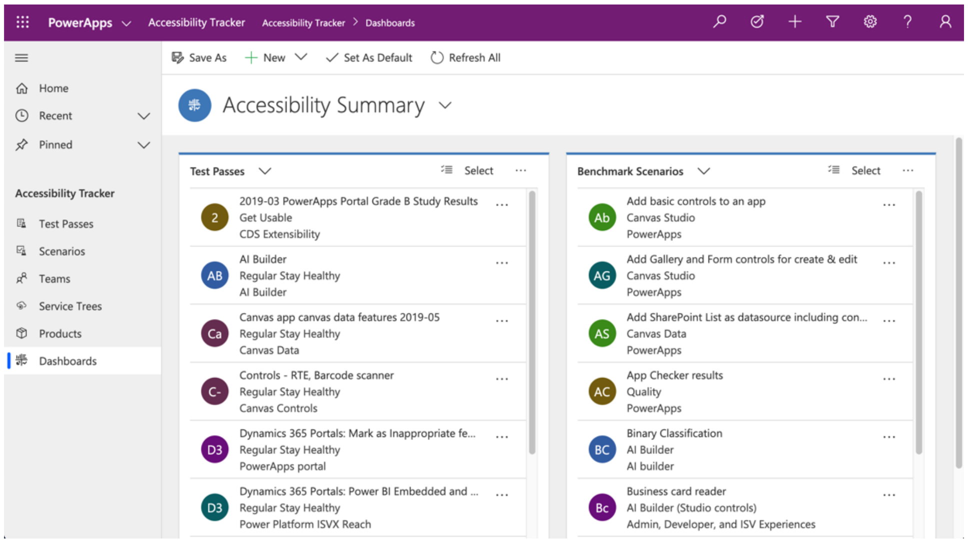
In Power Apps, there are three different types of apps: canvas, model-driven and portal. Each has a different set of capabilities to address different scenarios.

Canvas apps use connectors to access data and services. To create a UI, a maker can start with a blank screen like an artist's canvas and manually lay out each screen. This allows the creator to have complete control of the placement of controls on the canvas.



Example of a canvas app on multiple devices

Model-driven apps, on the other hand, require a Common Data Service database and are built on top of the data modeled in that database environment. Model-driven apps materialize views and detail screens based on the data structure. Because of this, they offer users a more consistent look and feel from one screen to the next without much effort by the creator.



Test Passes

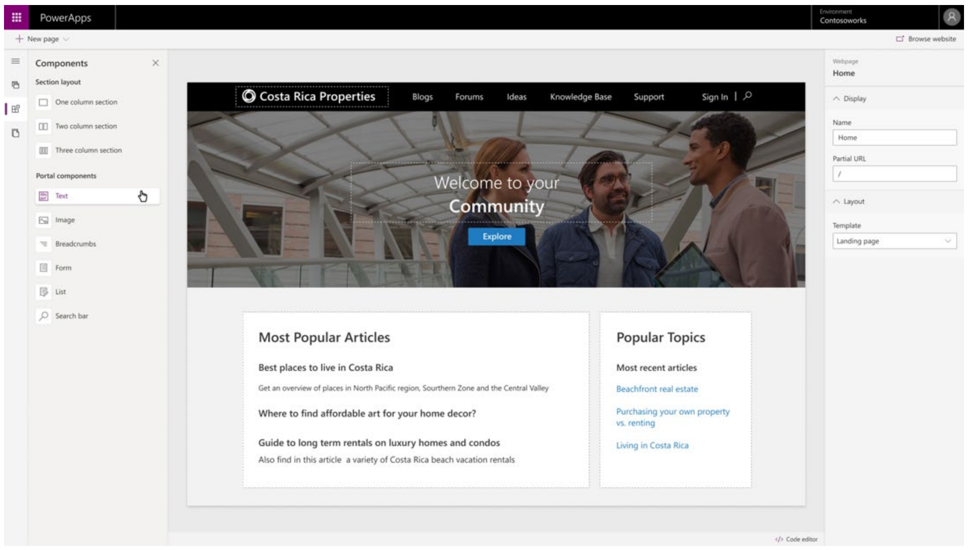
- 2019-03 PowerApps Portal Grade B Study Results / Get Usable / CDS Extensibility
- AI Builder / Regular Stay Healthy / AI Builder
- Canvas app canvas data features 2019-05 / Regular Stay Healthy / Canvas Data
- Controls - RTE, Barcode scanner / Regular Stay Healthy / Canvas Controls
- Dynamics 365 Portals: Mark as Inappropriate fe... / Regular Stay Healthy / PowerApps portal
- Dynamics 365 Portals: Power BI Embedded and ... / Regular Stay Healthy / Power Platform ISVX Reach

Benchmark Scenarios

- Add basic controls to an app / Canvas Studio / PowerApps
- Add Gallery and Form controls for create & edit / Canvas Studio / PowerApps
- Add SharePoint List as datasource including con... / Canvas Data / PowerApps
- App Checker results / Quality / PowerApps
- Binary Classification / AI Builder / AI builder
- Business card reader / AI Builder (Studio controls) / Admin, Developer, and ISV Experiences

Example of a model-driven app

Lastly, with portals, makers can expose the data they have in the Common Data Service to the broader world by creating a public facing website.



Most Popular Articles

Best places to live in Costa Rica

Get an overview of places in North Pacific region, Southern Zone and the Central Valley

Where to find affordable art for your home decor?

Guide to long term rentals on luxury homes and condos

Also find in this article a variety of Costa Rica beach vacation rentals

Popular Topics

Most recent articles

Beachfront real estate

Purchasing your own property vs. renting

Living in Costa Rica

Webpage: Home

Name: Home

Partial URL: /

Template: Landing page

Example of editing a Power Apps portal

Creating apps for Microsoft Teams

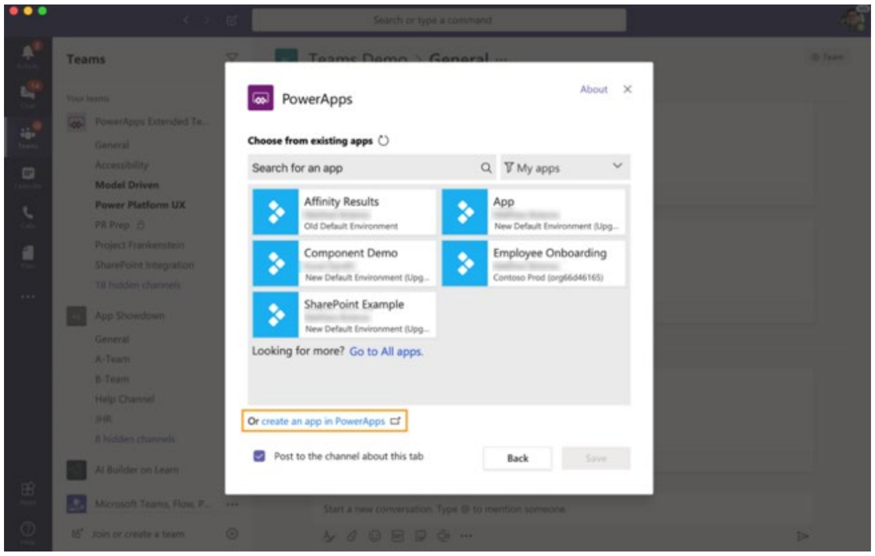
Getting started with customizing Microsoft Teams with Power Apps is simple. If you already have apps you've built using Power Apps, you can immediately start embedding them into Microsoft Teams. Otherwise, you can build a new app directly from Microsoft Teams using a Microsoft Teams template.

Get started with an optimized canvas app for Microsoft Teams

If you want to start building a net new canvas app for Microsoft Teams. The easiest way to get started is to leverage the optimized template for Microsoft Teams. Please note that this functionality will not be released until mid-November 2019.

To access the Microsoft Teams template:

1. Open Microsoft Teams.
2. Navigate to the channel you want to create an app for.
3. Select the + button to create a new tab.
4. Select **create an app in Power Apps.**


PowerApps
About
Choose from existing apps
Search for an app
My apps
Affinity Results
App
Component Demo
Employee Onboarding
SharePoint Example
Looking for more? Go to All apps.
Or create an app in PowerApps
Post to the channel about this tab
Back
Save

Create an app in Power Apps

This will open the canvas studio and default your new app with a few key settings:

- The theme is **Teams purple.**
- The screen size is tablet for Microsoft Teams desktop.
- And several preview optimizations are enabled by default to increase performance.

To further optimize your app, refer to the optimization section of this whitepaper. In that section, learn how to make your apps responsive and performant.

Once you are done creating your app, you can come back to the Power Apps modal within Microsoft Teams and select the refresh button before finally selecting the app you just built.

Adding an existing app into Microsoft Teams

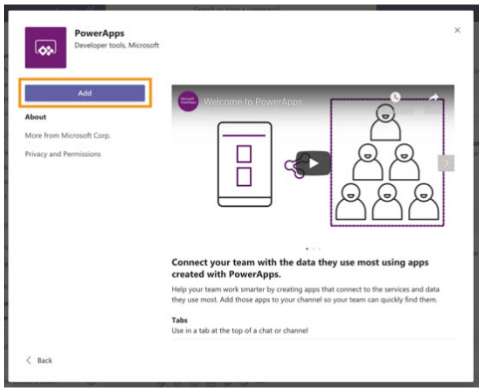
If you have already built an app within Power Apps, you can easily bring it into Microsoft Teams.

Canvas apps

With canvas apps, you can create pixel perfect applications that you can embed directly with Microsoft Teams. Canvas apps can be embedded both as tabs within Microsoft Teams channels and as personal apps. For inspiration, on how to leverage canvas apps within Microsoft Teams, check out the usage scenarios section of the white paper.

Pinning an app as a tab with the Power Apps app

1. Open Microsoft Teams.
2. Navigate to the channel or chat where you want to pin your Power Apps app.
3. Select **+** at the end of your tabs.
4. Search for “Power Apps.”
5. Select the **Power Apps** tile.
6. Select **Add**.



Connect your team with the data they use most using apps created with PowerApps.

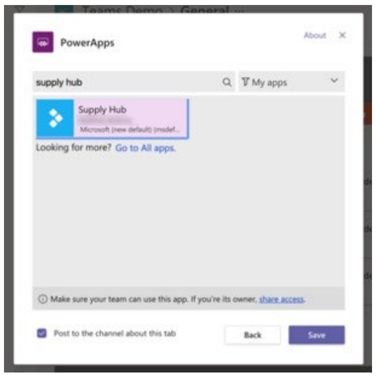
Help your team work smarter by creating apps that connect to the services and data they use most. Add those apps to your channel so your team can quickly find them.

Tabs

Use in a tab at the top of a chat or channel

Pinning an app as a tab with the Power Apps app

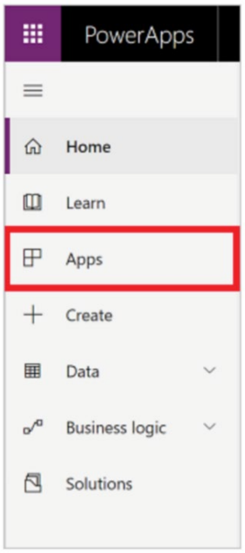
7. Select the app you want to pin.
8. Select **share access** to navigate to the sharing panel within Power Apps where you can share the app with the appropriate users.
9. Finally, come back to Microsoft Teams and select **Save** to pin the app.


PowerApps
About
supply hub
My apps
Supply Hub
Looking for more? Go to All apps.
Make sure your team can use this app. If you're its owner, share access.
Post to the channel about this tab
Back
Save

Save to pin the app

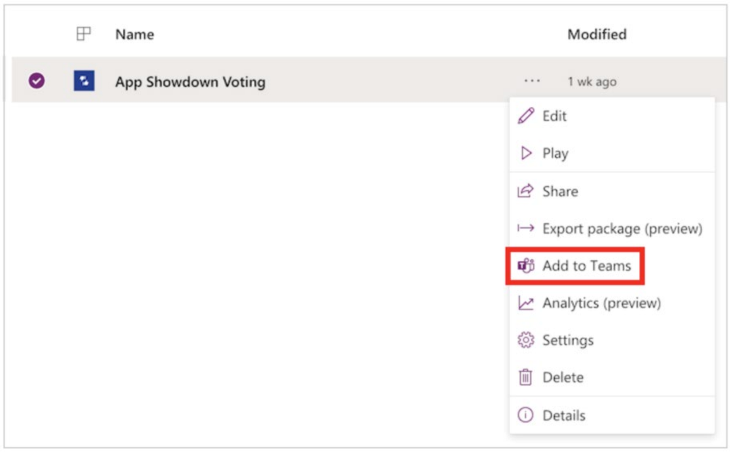
Adding a canvas app as a personal app

1. Sign in to make.powerapps.com.
2. Select **Apps** in the menu.



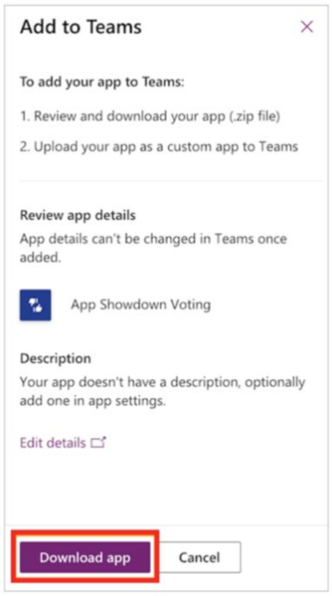
Adding a canvas app as a personal app

3. Select **More actions (...)** for the app you want to share in Microsoft Teams.
4. Select **Add to Teams**.



Add app to Teams

5. In the **Add to Teams** panel, select **Download**. Power Apps will then generate your Microsoft Teams manifest file using the app description and logo you've already set in your app.



Add to Teams

To add your app to Teams:

1. Review and download your app (.zip file)

2. Upload your app as a custom app to Teams

Review app details

App details can't be changed in Teams once added.

App Showdown Voting

Description

Your app doesn't have a description, optionally add one in app settings.

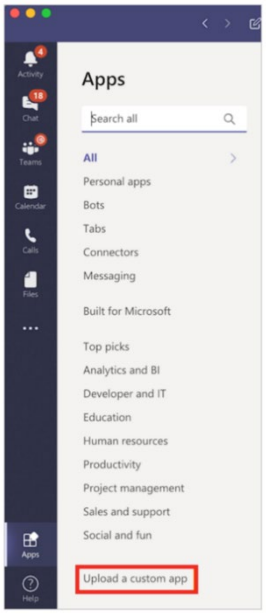
Edit details

Download app Cancel

Download app to generate your Microsoft Teams manifest file

6. Open Microsoft Teams.
7. Select **Apps** in the left navigation and then select **Upload a custom app**.

   The Upload a custom app only appears if your Microsoft Teams administrator has turned on **Allow uploading of custom apps**. Please refer to the administration section of this whitepaper to learn how to enable this functionality.



Upload a custom app

8. Select **Add** to add the app as a personal app.

Model-driven apps

By embedding model-driven apps into Microsoft Teams, you can have quick and easy access to any of your entities or lists within a Microsoft Teams channel. To use model-driven apps, you will want to take advantage of the Dynamics app within Microsoft Teams. A more feature complete embedding experience for model-driven apps will come later in 2019.

Note that the Dynamics collaboration feature is only enabled for a selected set of system entities:

- Account
- Agreement
- Annotation
- Appointment
- Businessunit
- Campaign
- Case
- Category
- Competitor
- Contact
- Email
- Expense

- Fulfillment Preference
- Inventory Adjustment
- Inventory Transfer
- Invoice
- Kbarticle
- Knowledge Article
- Lead
- Opportunity
- Opportunityproduct
- Order
- Product
- Project
- Purchase Order
- Purchase Order Receipt
- Quote
- Resource Request
- RMA
- RMA Receipt
- RTV
- Sales Literature
- Task
- Team
- Work Order

If you want to enable Microsoft Teams integration for additional entities or custom entities, you can do it programmatically using the msdyn_SetTeamsDocumentStatus Web API action.

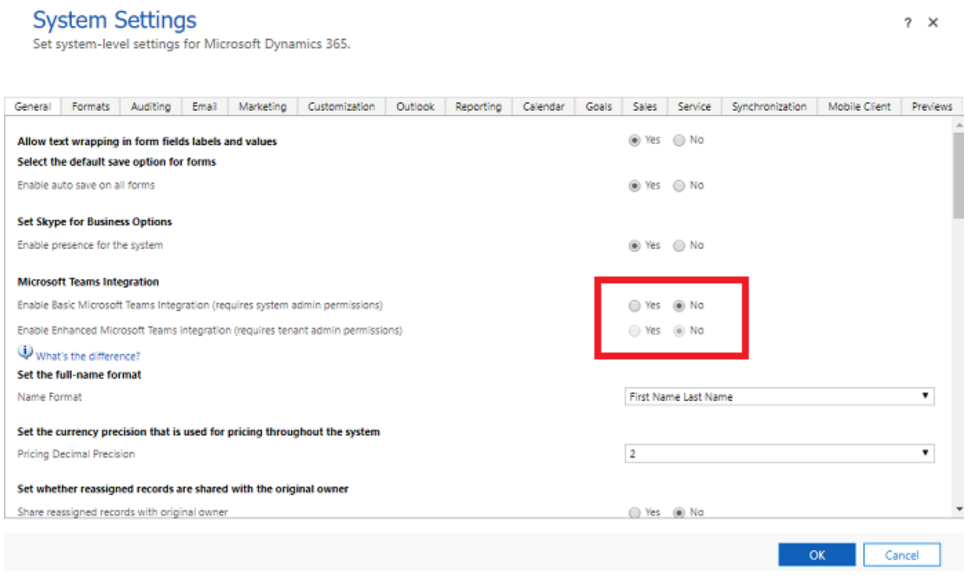
Enable Microsoft Teams integration in the Common Data Service

By default, the Basic and Enhanced Microsoft Teams Integration is disabled in the Common Data Service. To turn these features on, follow the steps below.

1. Sign in as a system administrator to Common Data Service.
2. Go to **Settings** > **Administration** > **System Settings** > **General** tab.

3. To enable basic collaboration experience, select **Yes** for **Enable Basic Microsoft Teams Integration**.

   When Basic Microsoft Teams Integration is enabled, the Collaborate button appears on records in model-driven apps in Dynamics 365 so you can see the connected team channel or set up a new connection in Microsoft Teams. In addition, in the Documents tab on model-driven app record page, the connected team channel file library will appear.



System Settings

Set system-level settings for Microsoft Dynamics 365.

General | Formats | Auditing | Email | Marketing | Customization | Outlook | Reporting | Calendar | Goals | Sales | Service | Synchronization | Mobile Client | Previews

Allow text wrapping in form fields labels and values [✓ Yes] No

Select the default save option for forms

Enable auto save on all forms [✓ Yes] No

Set Skype for Business Options

Enable presence for the system [✓ Yes] No

Microsoft Teams Integration

Enable Basic Microsoft Teams Integration (requires system admin permissions) Yes [✓ No]

Enable Enhanced Microsoft Teams Integration (requires tenant admin permissions) Yes [✓ No]

What's the difference?

Set the full-name format

Name Format: First Name Last Name

Set the currency precision that is used for pricing throughout the system

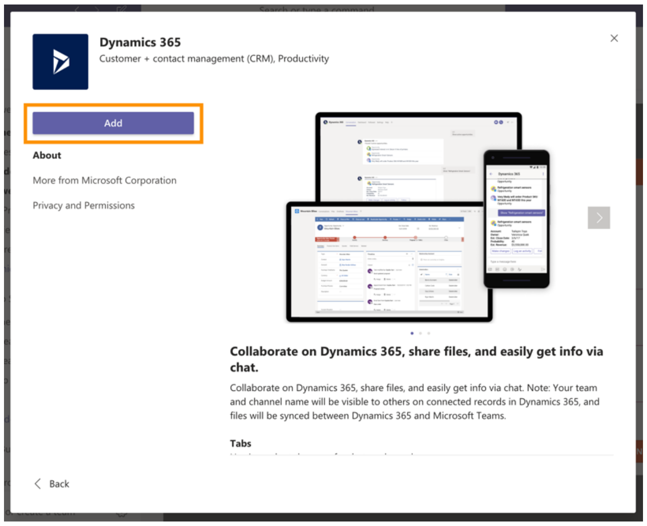
Pricing Decimal Precision: 2

Set whether reassigned records are shared with the original owner

Share reassigned records with original owner Yes [✓ No]

Add an app to a Microsoft Teams channel

1. Open Microsoft Teams.
2. Navigate to the channel or chat where you want to pin your Power Apps app.
3. Select **+** at the end of your tabs.
4. Search for “Dynamics 365.”
5. Select the **Dynamics 365** tile.
6. Select **Add**.



Dynamics 365

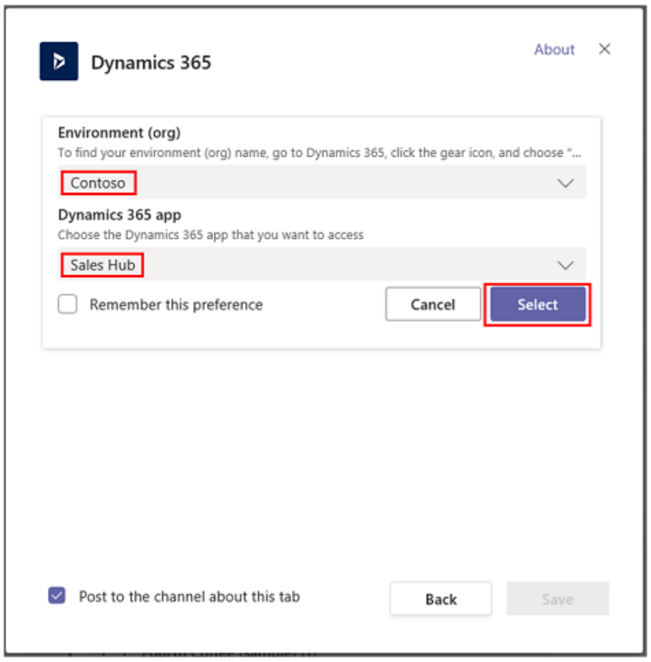
Customer + contact management (CRM), Productivity

Collaborate on Dynamics 365, share files, and easily get info via chat.

Collaborate on Dynamics 365, share files, and easily get info via chat. Note: Your team and channel name will be visible to others on connected records in Dynamics 365, and files will be synced between Dynamics 365 and Microsoft Teams.

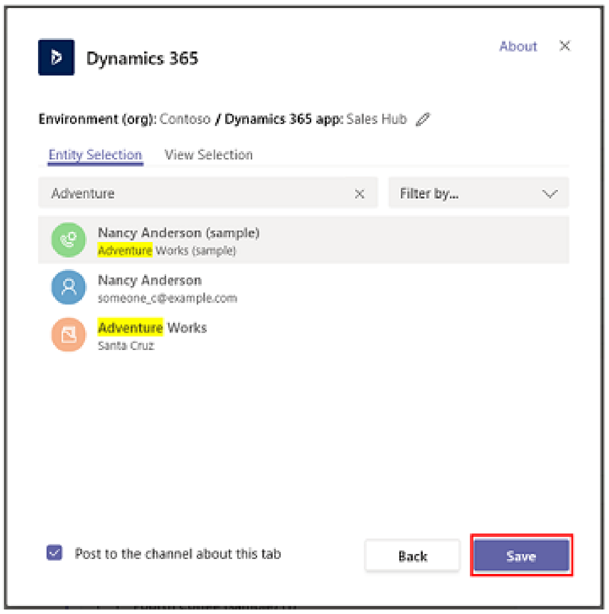
Tabs

7. Select a version 9.x environment and a Unified Interface app to connect, and then choose Save.

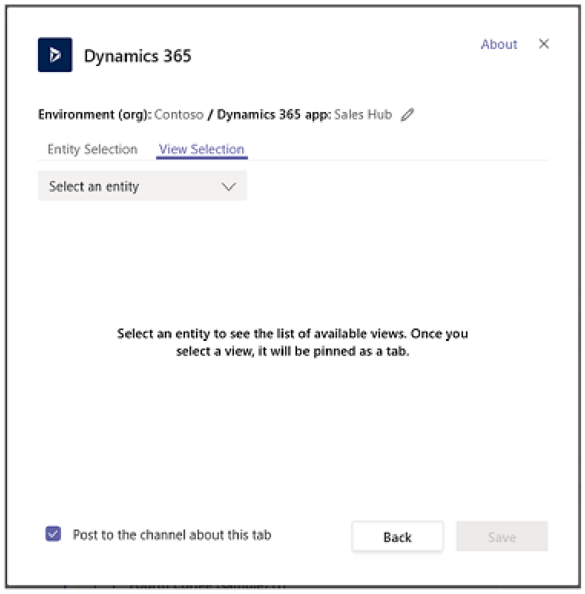

Dynamics 365
About
Environment (org)
To find your environment (org) name, go to Dynamics 365, click the gear icon, and choose "...
Contoso
Dynamics 365 app
Choose the Dynamics 365 app that you want to access
Sales Hub
Remember this preference
Cancel
Select
Post to the channel about this tab
Back
Save

Select an entity or view

To select an entity to connect. You can pick a recently viewed record or use search to find records. You can use Filter by to narrow the search to an entity type. Once you've picked a record, select Save.



To select a view, select an entity to see the list of available views. Once you've picked a view, select Save.


Dynamics 365
About
Environment (org): Contoso / Dynamics 365 app: Sales Hub
Entity Selection
View Selection
Select an entity
Post to the channel about this tab
Back
Save

Select an entity to see the list of available views. Once you select a view, it will be pinned as a tab.

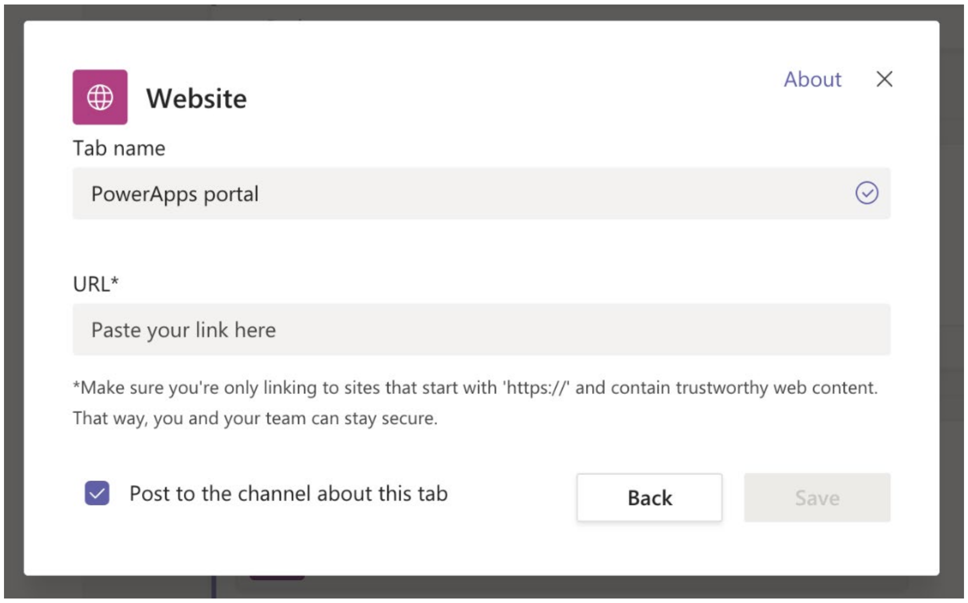
After completing the above steps, you will see a new Dynamics 365 tab in the selected team channel.

Portals

Power Apps portals are effectively websites, so to embed them within a Microsoft Teams channel, you can simply add them as a website tab.

Adding a portal as a website tab to Microsoft Teams

1. Open Microsoft Teams.
2. Navigate to the channel or chat where you want to pin your Power Apps app.
3. Select **+** at the end of your tabs.
4. Search for "Website."
5. Select the **Website** tile.
6. Provide a **Tab name** and the **URL** of your Power Apps portal



Tab name

PowerApps portal

URL*

Paste your link here

*Make sure you're only linking to sites that start with 'https://' and contain trustworthy web content. That way, you and your team can stay secure.

✓ Post to the channel about this tab

Adding a portal as a website tab to Microsoft Teams

7. Select **Save**.

Administering your apps within Microsoft Teams

If you are a Microsoft Teams admin, you have additional controls that allow you to distribute Power Apps within your tenant. This includes curating the app catalogue, automatically publishing personal apps to a subset of users, and controlling who can sideload applications into the tenant.

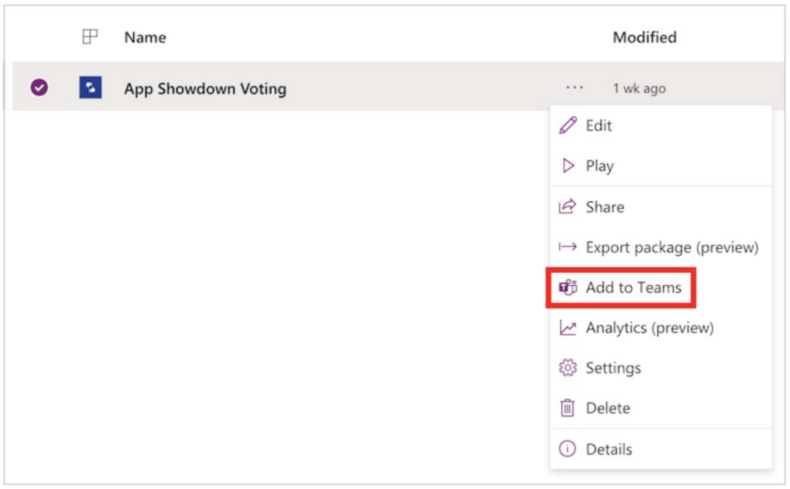
Adding apps to the app catalogue

You can use the Microsoft Teams Tenant Apps Catalog to test and distribute line-of-business applications to your organization. The Microsoft Teams Tenant Apps Catalog lets you distribute line-of-business applications that were built specifically for your organization and that you rely on to complete critical business functions.

To publish apps for your organization, sign into your Microsoft Teams client using an account with the global admin or teams service admin roles and then follow the instructions below.

Get a Microsoft Teams app package

Start by generating the app package for the app you want to publish to the catalogue. For canvas apps, simply export your app for Microsoft Teams from Power Apps.



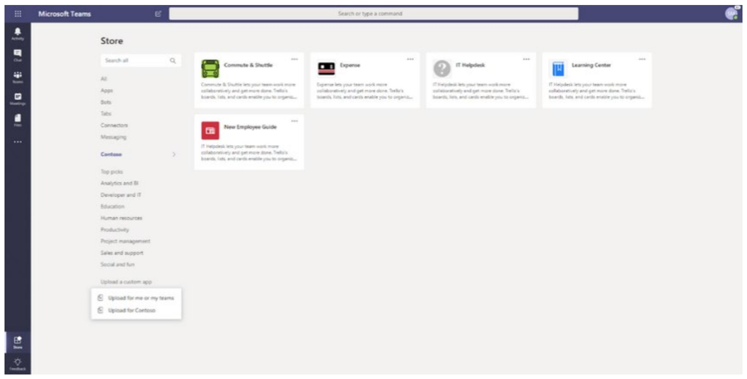
Get a Microsoft Teams app package

For other types of apps, you can use App Studio to create the package. Instructions on how to use app studio are in the advanced topics section of this whitepaper.

Once you have the app package, you can add it to the enterprise app catalog. While all users in the tenant can view the app catalog, only global admins and teams service admins can publish and manage it.

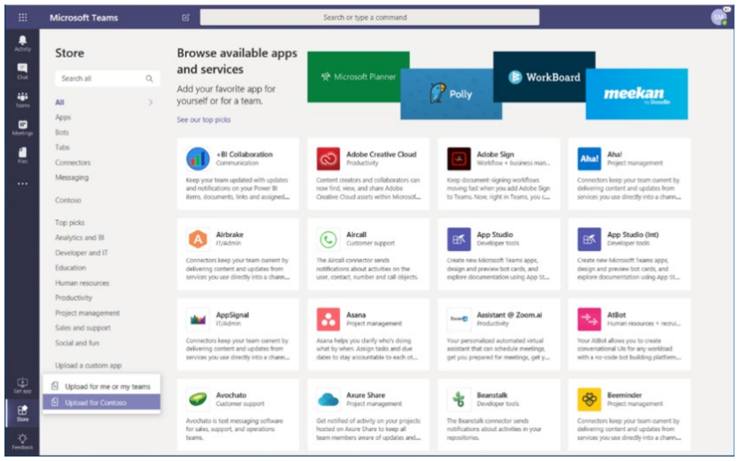
Go to the Tenant Apps Catalogue

Start the Microsoft Teams client and sign in using your global or teams service admin credentials. From the Microsoft Teams Store, select the new section named for your specific organization. In this example, the tenant name is Contoso, while you are using Microsoft Teams, it will be replaced with the name of your company. Users in your organization can view apps in the catalog and install them for teams of which they are a member.

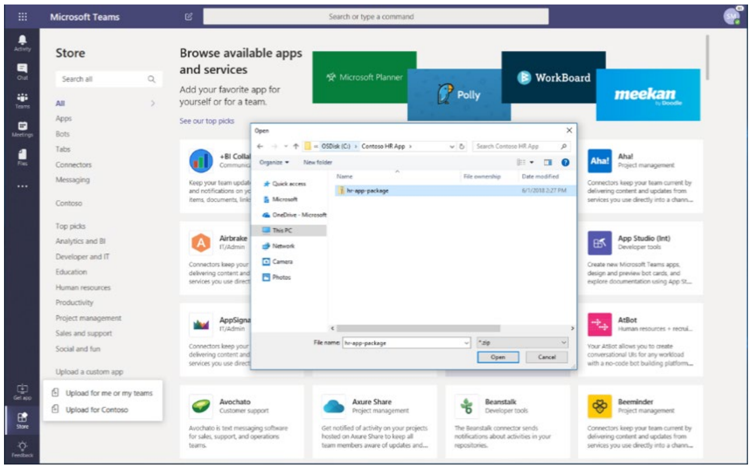


Add an app to the Tenant Apps Catalogue

From the store, select **Upload a custom app** > **Upload for Contoso**.



Navigate to the app package and select it, and then click Open.

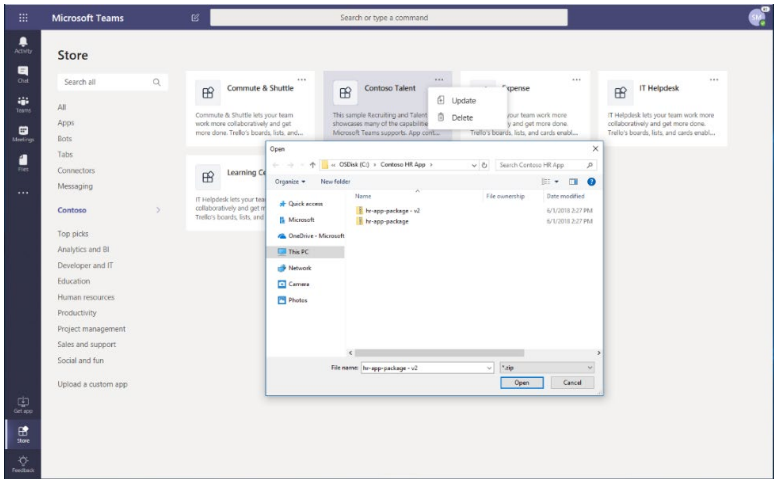

Microsoft Teams
Store
Browse available apps and services
Add your favorite app for yourself or for a team.
Upload for me or my teams
Upload for Contoso
Open
File name: hr-app-package
Open
Cancel

When you go back to your Tenant Apps Catalog, the new enterprise app will be there. Remember, only you and members of your organization have access to this app catalog.

Update an app in the Tenant Apps Catalogue

From your Tenant Apps Catalog, select "…" on the top right of the app you want to update.

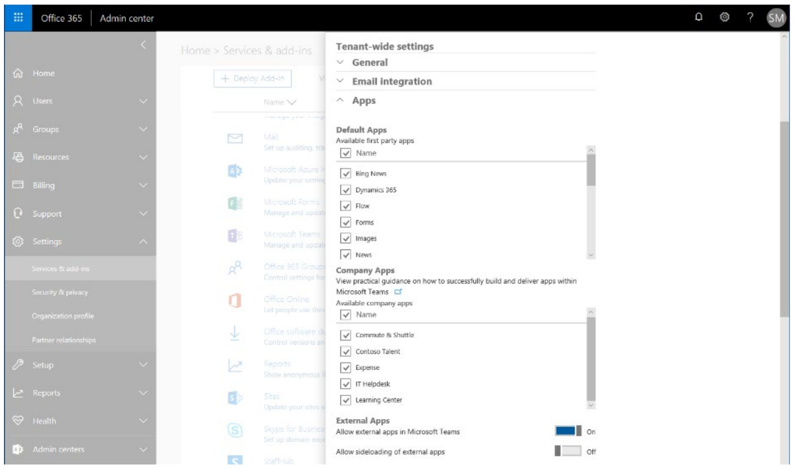
Navigate to the updated app package and select it, and then click Open.



The app will be revised to version 2.0. You can also delete the app for your entire company from this menu.

Use the Office 365 admin portal to manage the Tenant Apps Catalogue

If you have apps that need bug fixes, you can temporarily disable apps through the Office 365 admin portal. Select **Settings** > **Services & add-ins** > **Microsoft Teams**. In addition to previous settings, there is now a section dedicated to your company's apps. You can choose which apps you want to enable or disable.



Tenant-wide settings

- General
- Email integration
- Apps

Default Apps
Available first party apps

- [✓] Name
- [✓] Bing News
- [✓] Dynamics 365
- [✓] Flow
- [✓] Forms
- [✓] Images
- [✓] News

Company Apps
View practical guidance on how to successfully build and deliver apps within Microsoft Teams

Available company apps

- [✓] Name
- [✓] Commute & Shuttle
- [✓] Contoso Talent
- [✓] Expense
- [✓] IT Helpdesk
- [✓] Learning Center

External Apps
Allow external apps in Microsoft Teams: On
Allow sideloading of external apps: Off

Disabling an app will block users from interacting with the app, without deleting the app entirely. These controls give you additional flexibility and control when managing apps in your enterprise.

Allowing custom apps in your tenant

As an admin, you can use custom app policies and settings to control who in your organization can upload custom apps to Microsoft Teams. Admins decide which users can upload custom apps, and admins and team owners can determine whether specific teams in your organization allow custom apps to be added to them.

For end-users within your tenant to upload customized personal apps from Power Apps, you will need to enable uploading of custom apps for your tenant.

Custom app policy and settings

Three components determine whether a user can upload a custom app to a team, giving you granular control over who can add custom apps to a team and which teams custom apps can be added to:

- User custom app policy
- Team custom app setting
- Org-wide custom app setting

User custom app policy

As part of app setup policies, admins can use a policy setting, **Allow uploading custom apps**, to control whether a user can upload custom apps to Microsoft Teams.

If this setting is turned off:

- The user can't upload a custom app to any team in your organization or in the personal context.
- The user can interact with custom apps, depending on the org-wide custom app setting.

If this setting is turned on:

- The user can upload custom apps to teams that allow it and to teams for which they are owners, depending on the org-wide custom app setting.
- The user can upload custom apps to the personal context.
- The user can interact with custom apps, depending on the org-wide custom app setting.

You can edit the settings in the global app setup policy to include the apps that you want. If you want to customize Microsoft Teams for different groups of users in your organization, create and assign one or more custom app setup policies.

To set a user custom app policy:

1. In the left navigation of the Microsoft Teams admin center, go to **Teams apps** > **Setup policies**.
2. Click **Add**.
3. Turn on or turn off **Allow uploading custom apps**.
4. Choose any other settings that you want to for the policy.
5. Click Save.

Team custom app setting

Admins and team owners can control whether a team allows for custom apps to be added to it. This setting, **Allow members to upload custom apps**, together with a user's custom app policy determines who can add custom apps to a team.

If this setting is turned off:

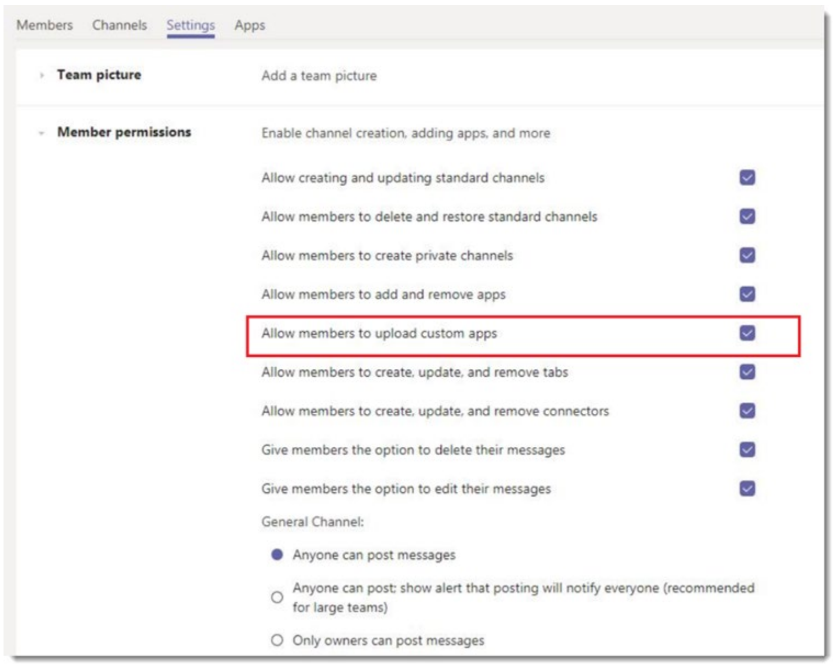
- Team owners can add custom apps, if their custom app policy allows it.
- Team members who aren't team owners can't add custom apps to the team.

If this setting is turned on:

- Team owners can add custom apps, if their custom app policy allows for it.
- Team members who aren't team owners can add custom apps, if their custom app policy allows for it.

To configure the team custom app setting:

1. In Microsoft Teams, go to the team, click **More options ...** > **Manage team**.
2. Select **Settings**, and then expand **Member permissions**.
3. Select or clear the **Allow members to upload custom apps** check box.


Members Channels Settings Apps

Team picture — Add a team picture

Member permissions — Enable channel creation, adding apps, and more

- Allow creating and updating standard channels ✓
- Allow members to delete and restore standard channels ✓
- Allow members to create private channels ✓
- Allow members to add and remove apps ✓
- Allow members to upload custom apps ✓
- Allow members to create, update, and remove tabs ✓
- Allow members to create, update, and remove connectors ✓
- Give members the option to delete their messages ✓
- Give members the option to edit their messages ✓

General Channel:

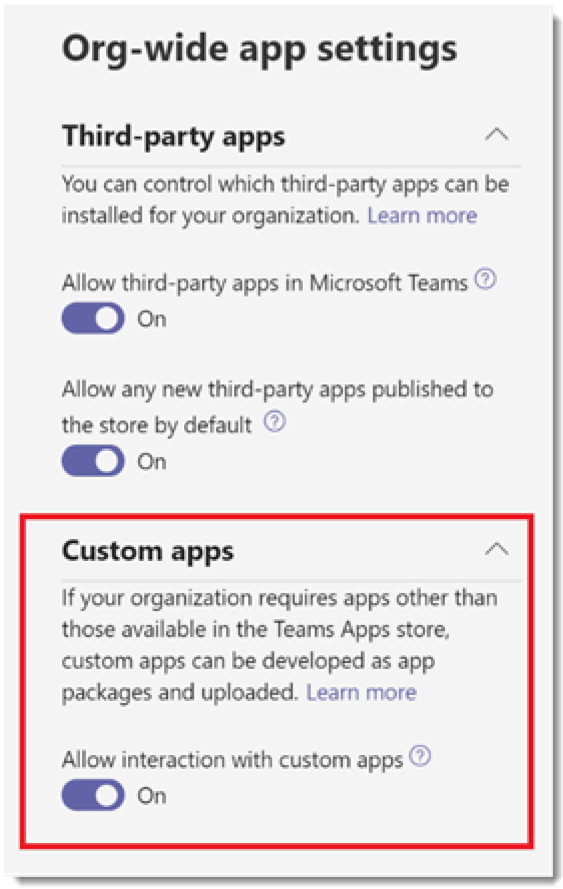
- ● Anyone can post messages
- ○ Anyone can post; show alert that posting will notify everyone (recommended for large teams)
- ○ Only owners can post messages

Org-wide custom app setting

The org-wide custom app setting, **Allow interaction with custom apps**, applies to everyone in your organization and governs whether they can upload or interact with custom apps. This setting overrides the user and team custom app policy and setting. It's intended to serve as a master on/off switch during security events.

To configure the org-wide custom app setting:

1. In the left navigation of the Microsoft Teams admin center, go to **Teams apps** > **Permission policies**.
2. Select **Org-wide app settings**.
3. Under **Custom apps**, turn on or turn off **Allow interaction with custom apps**.


Org-wide app settings

Third-party apps

You can control which third-party apps can be installed for your organization. Learn more

Allow third-party apps in Microsoft Teams
On

Allow any new third-party apps published to the store by default
On

Custom apps

If your organization requires apps other than those available in the Teams Apps store, custom apps can be developed as app packages and uploaded. Learn more

Allow interaction with custom apps
On

How custom app policies and settings work together

This table summarizes the custom app policy and settings, how they work together, and their combined effect on controlling who in your organization can upload custom apps to Microsoft Teams. Say, for example, you want to allow only team owners to upload custom apps to specific teams. You would set the following:

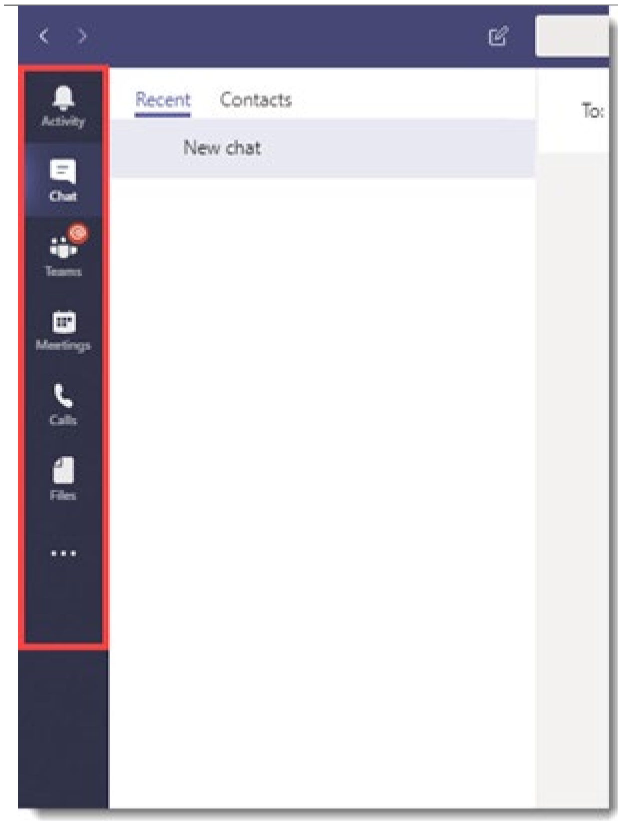
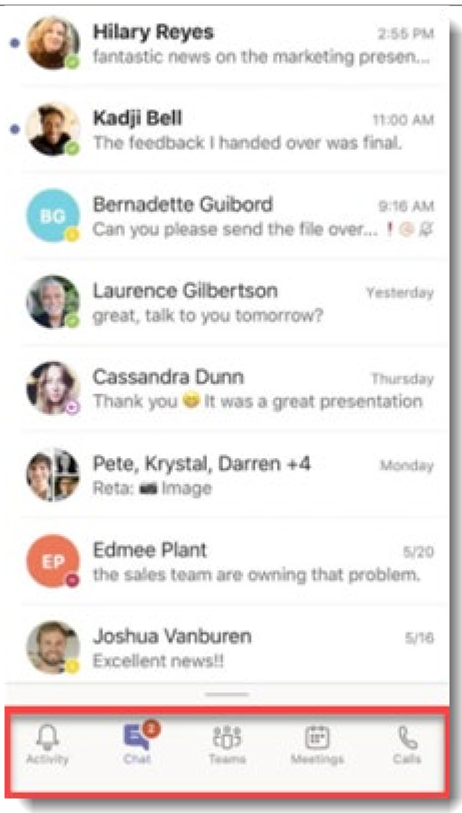
- Turn on the **Allow interaction with custom apps** setting in the Microsoft Teams admin center.
- Turn off the **Allow members to upload custom apps** for every team to which you want to restrict access.
- Create and assign a custom app setup policy in the Microsoft Teams admin center with the **User can upload custom apps** setting turned on and assign it to the team owners.

Org-wide custom app setting	Team custom app setting	User custom app policy	Effect
Off	Off	Off	Interaction with all custom apps is blocked for your organization. Custom apps can't be uploaded by anyone. You can use PowerShell to remove the custom app.
Off	Off	On	Interaction with all custom apps is blocked for your organization. Custom apps can't be uploaded by anyone. You can use PowerShell to remove the custom app.
Off	On	Off	Interaction with all custom apps is blocked for your organization. Custom apps can't be uploaded by anyone. You can use Windows PowerShell to delete custom apps.
Off	On	On	Interaction with all custom apps is blocked for your organization. Custom apps can't be uploaded by anyone. You can use PowerShell to remove the custom app.
On	Off	Off	The user can't upload custom apps.
On	Off	On	If the user is a team owner, they can upload custom apps to the team. If the user isn't a team owner, they can't upload custom apps to the team. The user can upload custom apps in the personal context.
On	On	Off	The user can't upload custom apps.
On	On	On	The user can upload custom apps to the team, regardless of whether the user is a team owner. The user can upload custom apps in the personal context.

Pinning apps

As an admin, you can use app setup policies to customize Microsoft Teams to highlight the apps that are most important for your users. You choose the apps to pin and set the order that they appear. App setup policies let you showcase apps that users in your organization need, including those built by third parties or by developers in your organization. You can also use app setup policies to manage how built-in features appear.

Apps are pinned to the app bar. This is the bar on the side of the Microsoft Teams desktop client and at the bottom of the Microsoft Teams mobile clients (iOS and Android).

Microsoft Teams desktop client	Microsoft Teams mobile client
	

Hilary Reyes 2:55 PM
fantastic news on the marketing presen...

Kadji Bell 11:00 AM
The feedback I handed over was final.

Bernadette Guibord 9:16 AM
Can you please send the file over...

Laurence Gilbertson Yesterday
great, talk to you tomorrow?

Cassandra Dunn Thursday
Thank you 😊 It was a great presentation

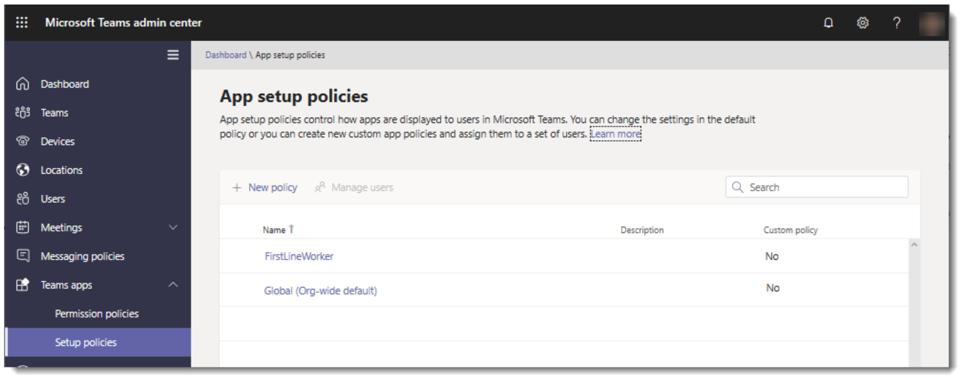
Pete, Krystal, Darren +4 Monday
Reta: Image

Edmee Plant 5/20
the sales team are owning that problem.

Joshua Vanburen 5/16
Excellent news!!

You manage app setup policies in the Microsoft Teams admin center. You can use the global (Org-wide default) policy or create custom policies and assign them to users. Users in your organization will automatically get the global policy unless you create and assign a custom policy.

You can edit the settings in the global policy to include the apps that you want. If you want to customize Microsoft Teams for different groups of users in your organization, create and assign one or more custom policies. If a user is assigned a custom policy, that policy applies to the user. If a user isn't assigned a custom policy, the global policy applies to the user.



App setup policies

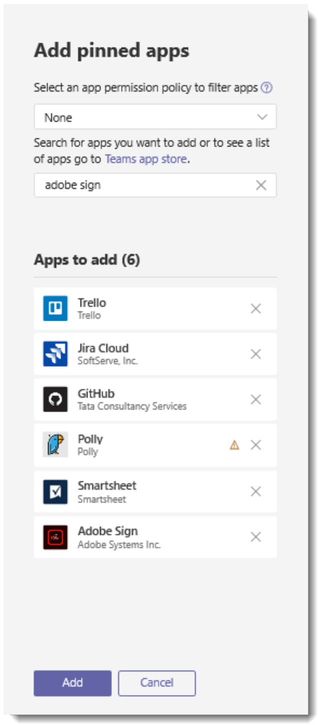
App setup policies control how apps are displayed to users in Microsoft Teams. You can change the settings in the default policy or you can create new custom app policies and assign them to a set of users. Learn more

Name ↑	Description	Custom policy
FirstLineWorker		No
Global (Org-wide default)		No

Create a custom app setup policy

You can use the Microsoft Teams admin center to create a custom policy.

1. In the left navigation of the Microsoft Teams admin center, go to **Teams apps** > **Setup policies**.
2. Select **Add**.
3. Enter a name and description for the policy, and then select **Add apps**.
4. Turn on or turn off **Allow uploading custom apps**, depending on whether you want to let users upload custom apps to Microsoft Teams. You won't be able to change this setting if **Allow third-party or custom apps** is turned off in org-wide app settings in app permission policies.
5. In the **Add pinned apps** pane, search for the apps you want to add, and then select **Add**. You can also filter apps by app permission policy. When you've chosen your list of apps, select **Add**.



Add pinned apps

Select an app permission policy to filter apps

None

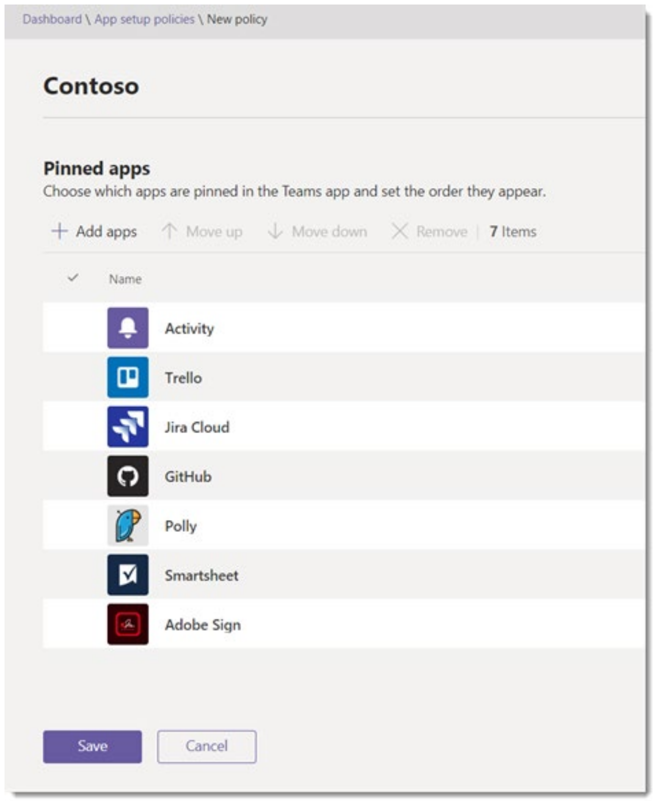
Search for apps you want to add or to see a list of apps go to Teams app store.

adobe sign

Apps to add (6)

Trello	Trello
Jira Cloud	SoftServe, Inc.
GitHub	Tata Consultancy Services
Polly	Polly
Smartsheet	Smartsheet
Adobe Sign	Adobe Systems Inc.

Arrange the apps in the order that you want them to appear in Microsoft Teams, and then select **Save**.


Dashboard \ App setup policies \ New policy

Contoso

Pinned apps

Choose which apps are pinned in the Teams app and set the order they appear.

Activity
Trello
Jira Cloud
GitHub
Polly
Smartsheet
Adobe Sign

Edit an app setup policy

You can use the Microsoft Teams admin center to edit a policy, including the global (Org-wide default) policy and custom policies that you create.

1. In the left navigation of the Microsoft Teams admin center, go to **Teams apps** > **Setup policies**.
2. Select the policy by clicking to the left of the policy name, and then select **Edit**.
3. From here, make the changes that you want. You can add, remove, and change the order of apps.
4. Select **Save**.

Assign a custom app setup policy to users

You can use the Microsoft Teams admin center to assign a custom policy to individual users or the Skype for Business PowerShell module to assign a custom policy to groups of users, such as a security group or distribution group.

1. In the left navigation of the Microsoft Teams admin center, go to **Users**.
2. Select the user by selecting to the left of the username, and then select **Edit settings**.
3. Under **App setup policy**, select the app setup policy you want to assign, and then select **Apply**.

To assign a policy to multiple users at a time:

1. In the left navigation of the Microsoft Teams admin center, go to **Teams apps** > **Setup policies**.
2. Select the policy by clicking to the left of the policy name.
3. Select **Manage users**.
4. In the **Manage users** pane, search for the user by display name or by username, select the name, and then select **Add**. Repeat this step for each user that you want to add.
5. When you're finished adding users, select **Save**.
6. Assign a custom app setup policy to users in a group

Optimizing your canvas apps for Microsoft Teams

To make your app feel even more native within Microsoft Teams, there are several additional optimizations that you can make so that it looks great and feels fast.

Responsiveness

Microsoft Teams works great on both desktop and mobile. To make sure that your app also works great on both screen sizes, you will want to make your app responsive. If you create a responsive layout, controls can respond to different devices or window sizes, making various experiences feel more natural. This will also remove the gray bars from your app whenever you embed it inside of Microsoft Teams.

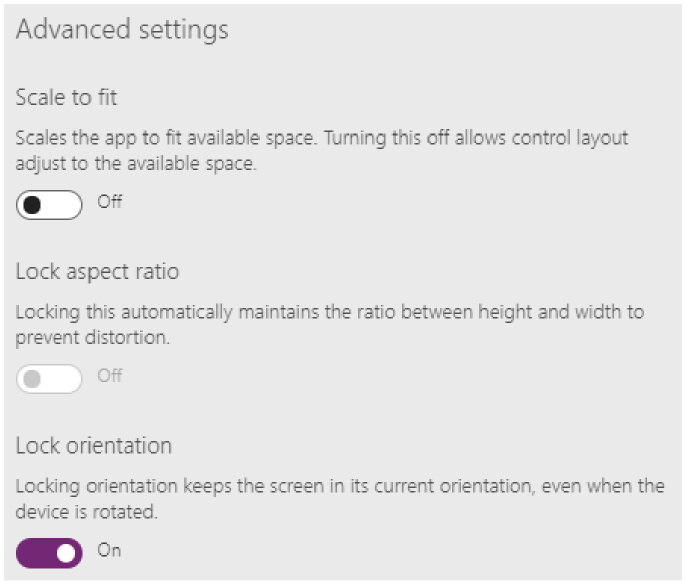
To achieve responsive layout, you must first start with the tablet layout, adjust some settings, and write a few expressions throughout your app so that controls can adjust to a range of screen sizes.

The following content is a subset of the content found on the full doc article on how to create responsive Power Apps app. To view the article in its entirety, visit aka.ms/responsivePowerApps.

Disable Scale to fit

The first step in enabling responsiveness is to configure each screen so that its layout adapts to the actual space in which the app is running.

To achieve this, navigate to **File** > **App settings** > **Screen size + orientation** and turn off the app's **Scale to fit** setting, which is on by default. When you turn this setting off, you also turn off Lock aspect ratio because you're no longer designing for a specific screen orientation.


Advanced settings

Scale to fit

Scales the app to fit available space. Turning this off allows control layout adjust to the available space.

Off

Lock aspect ratio

Locking this automatically maintains the ratio between height and width to prevent distortion.

Off

Lock orientation

Locking orientation keeps the screen in its current orientation, even when the device is rotated.

On

Understand app and screen dimensions

To make your app's layouts respond to changes in the screen dimensions, you'll write formulas that use the **Width** and **Height** properties of the screen. To show these properties, open an app in Power Apps Studio, and then select a screen. The default formulas for these properties appear on the Advanced tab of the right-hand pane.

```
Width = Max(App.Width, App.DesignWidth)

Height = Max(App.Height, App.DesignHeight)
```

These formulas refer to the **Width**, **Height**, **DesignWidth**, and **DesignHeight** properties of the app. The app's **Width** and **Height** properties correspond to the dimensions of the device or browser window in which your app is running. If the user resizes the browser window (or rotates the device if you've turned off **Lock orientation**), the values of these properties change dynamically. The formulas in the screen's **Width** and **Height** properties are reevaluated when these values change.

The **DesignWidth** and **DesignHeight** properties come from the dimensions that you specify in the Screen size + orientation pane of App settings. For example, if you select the tablet layout in landscape orientation, **DesignWidth** is 1366, and **DesignHeight** is 768.

Since they're used in the formulas for the screen's **Width** and **Height** properties, you can think of **DesignWidth** and **DesignHeight** as the minimum dimensions for which you'll design the app. If the actual area available to your app is even smaller than these minimum dimensions, the formulas for the screen's **Width** and **Height** properties ensure that their values won't become any smaller than these minimums. In that case, the user must scroll to view all of the screen's content.

If you want to build an app that looks great on both the Microsoft Teams desktop app and mobile app, it is recommended that you change the **Width** and **Height** formulas to the following for each screen. This will ensure that a user never sees scroll bars.

```
Width = App.Width

Height = App.Height
```

Use formulas for dynamic layout

To create a responsive design, you locate and size each control by using formulas instead of absolute (constant) coordinate values. These formulas express each control's position and size in terms of the overall screen size or relative to other controls on the screen.

> **Important**
>
> After you write formulas for the X, Y, Width and Height properties of a control, your formulas will be overwritten with constant values if you subsequently drag the control in the canvas editor. When you start to use formulas to achieve dynamic layout, you should avoid dragging controls.

In the simplest case, one control fills an entire screen. To create this effect, set the control's properties to these values:

Property	Value
X	`0`
Y	`0`
Width	`Parent.Width`
Height	`Parent.Height`

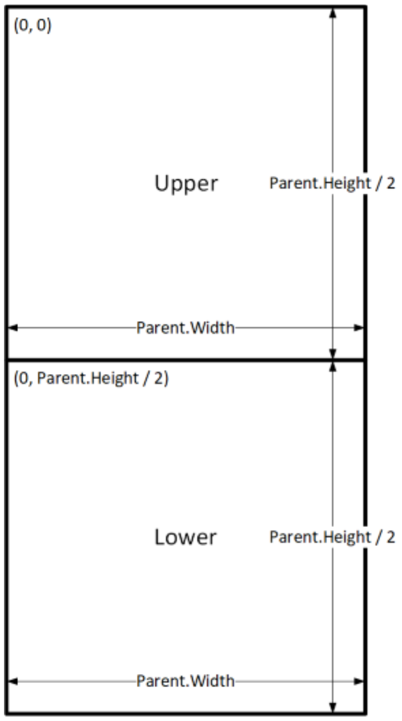
These formulas use the Parent operator. For a control placed directly on a screen, Parent refers to the screen. With these property values, the control appears in the upper-left corner of the screen (0, 0) and has the same Width and Height as the screen.

Later in this topic, you'll apply these principles (and the Parent operator) to position controls inside other containers, such as galleries, group controls, and components.

As an alternative, the control can fill only the top half of the screen. To create this effect, set the Height property to Parent.Height / 2, and leave the other formulas unchanged.

If you want a second control to fill the bottom half of the same screen, you can take at least two other approaches to constructing its formulas. For simplicity, you might take this approach:

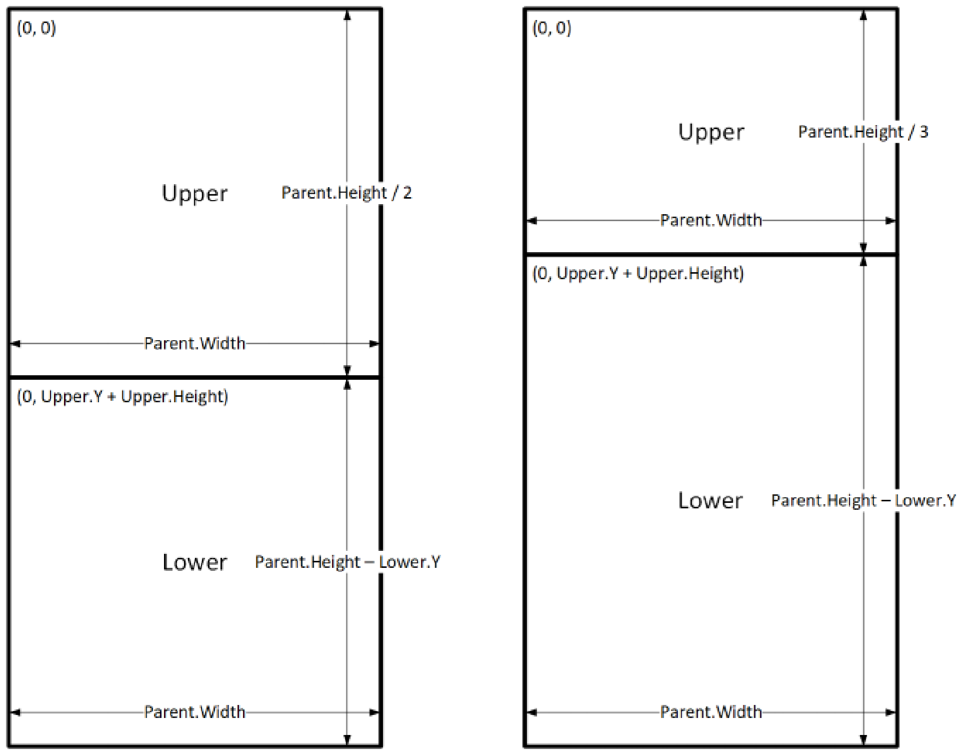
Control	Property	Formula
Upper	**X**	`0`
Upper	**Y**	`0`
Upper	**Width**	`Parent.Width`
Upper	**Height**	`Parent.Height / 2`
Lower	**X**	`0`
Lower	**Y**	`Parent.Height / 2`
Lower	**Width**	`Parent.Width`
Lower	**Height**	`Parent.Height / 2`



This configuration would achieve the effect that you want, but you'd need to edit each formula if you changed your mind about the relative sizes of the controls. For example, you might decide that the top control should occupy only the top one-third of the screen, with the bottom control filling the lower two-thirds.

To create that effect, you'd need to update the Height property of the Upper control and the Y and Height properties of the Lower control. Instead, consider writing the formulas for the Lower control in terms of the Upper control (and itself), as in this example:

Control	Property	Formula
Upper	**X**	`0`
Upper	**Y**	`0`
Upper	**Width**	`Parent.Width`
Upper	**Height**	`Parent.Height / 2`
Lower	**X**	`0`
Lower	**Y**	`Upper.Y + Upper.Height`
Lower	**Width**	`Parent.Width`
Lower	**Height**	`Parent.Height - Lower.Y`



With these formulas in place, you need only change the Height property of the Upper control to express a different fraction of the height of the screen. The Lower control automatically moves and resizes to account for the change.

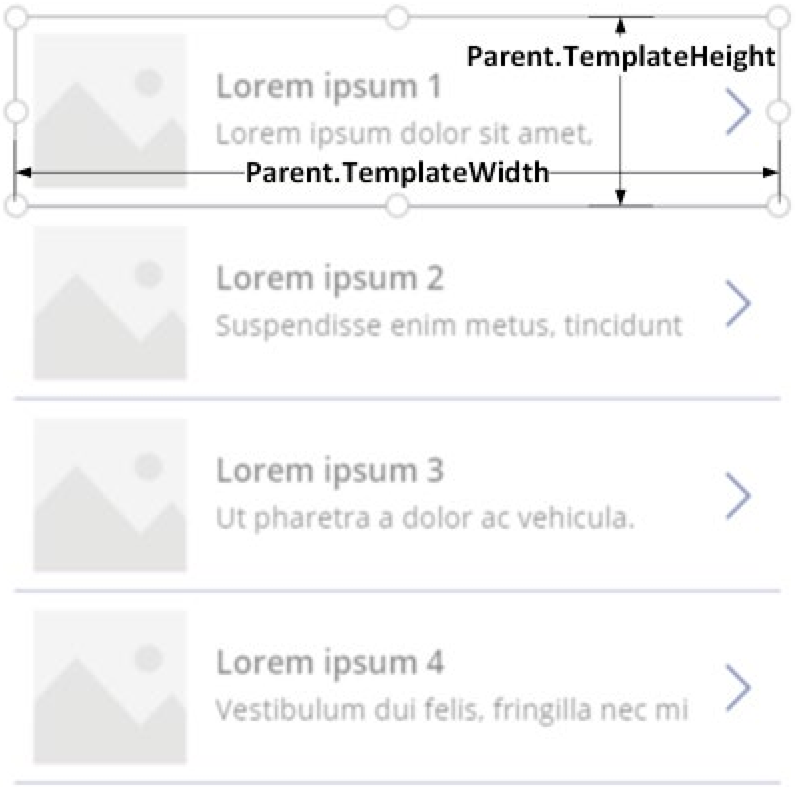
For more examples on how to responsively layout your screen please refer to the full documentation on responsive layout: aka.ms/responsivePowerApps.

Hierarchical layout

As you construct screens that contain more controls, it will become more convenient (or even necessary) to position controls relative to a parent control, rather than relative to the screen or a sibling control. By organizing your controls into a hierarchical structure, you can make your formulas easier to write and maintain.

Galleries

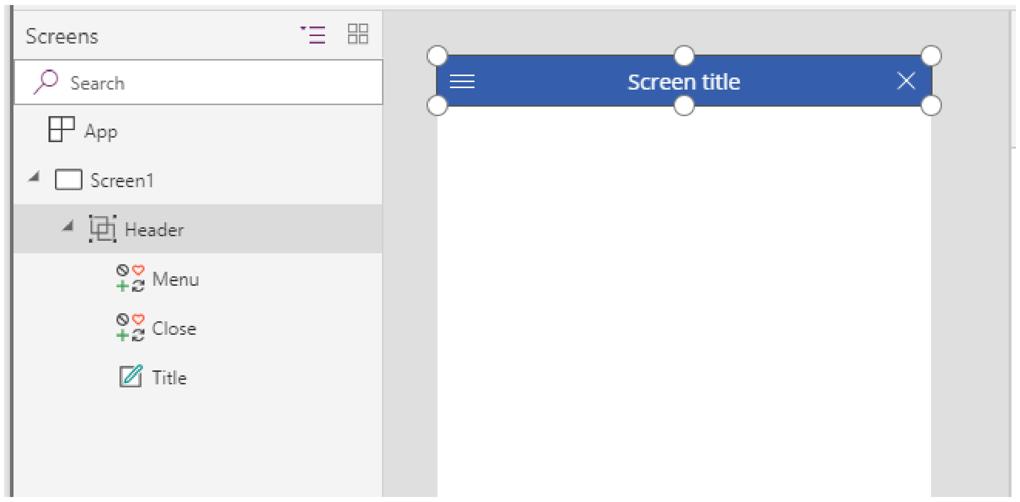
If you use a gallery in your app, you'll need to lay out controls within the gallery's template. You can position these controls by writing formulas that use the Parent operator, which will refer to the gallery template. In the formulas on controls within a gallery template, use the **Parent.TemplateHeight** and **Parent.TemplateWidth** properties; don't use **Parent.Width** and **Parent.Height**, which refer to the overall size of the gallery.


Parent.TemplateHeight
Lorem ipsum 1
Lorem ipsum dolor sit amet.
Parent.TemplateWidth
Lorem ipsum 2
Suspendisse enim metus, tincidunt
Lorem ipsum 3
Ut pharetra a dolor ac vehicula.
Lorem ipsum 4
Vestibulum dui felis, fringilla nec mi

Container control

You can use an experimental feature, the Container control, as a parent control. To turn this feature on, select **File** > **App settings** > **Advanced settings** and enable **Container control**.

Consider the example of a header at the top of a screen. It's common to have a header with a title and several icons with which your users can interact. You can construct such a header using the Container control, containing a Label control and two Icon controls:



Set the properties for these controls to these values:

Property	**Header**	**Menu**
X	`0`	`0`
Y	`0`	`0`
Width	`Parent.Width`	`Parent.Height`
Height	`64`	`Parent.Height`

Property	**Close**	**Title**
X	`Parent.Width - Close.Width`	`Menu.X + Menu.Width`
Y	`0`	`0`
Width	`Parent.Height`	`Close.X - Title.X`
Height	`Parent.Height`	`Parent.Height`

For the **Header** control, Parent refers to the screen. For the others, Parent refers to the Header control.

Having written these formulas, you can adjust the size or position of the Header control by changing the formulas for its properties. The sizes and positions of the child controls will automatically adjust accordingly.

Components

If you use another experimental feature, named Components, you can construct building blocks and reuse them throughout your app. As with the Container control, the controls that you place within a component should base their position and size formulas on **Parent.Width** and **Parent.Height**, which refer to the size of the component.

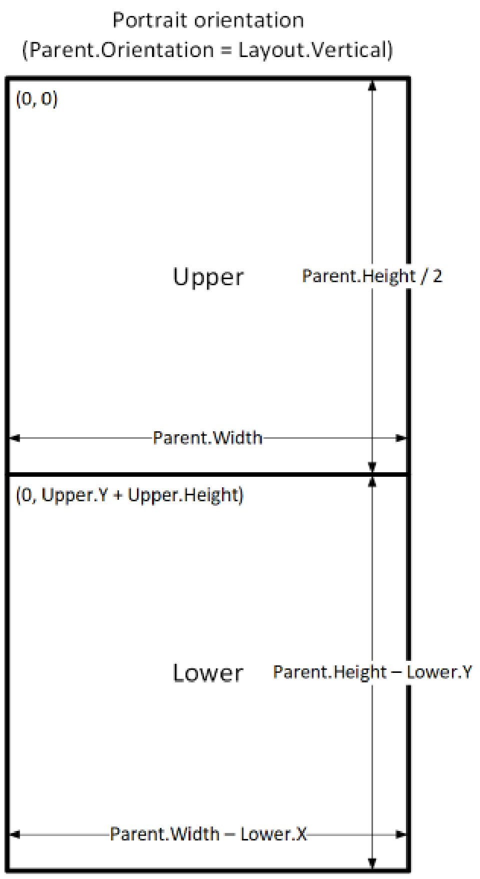
Adapting layout for device size and orientation

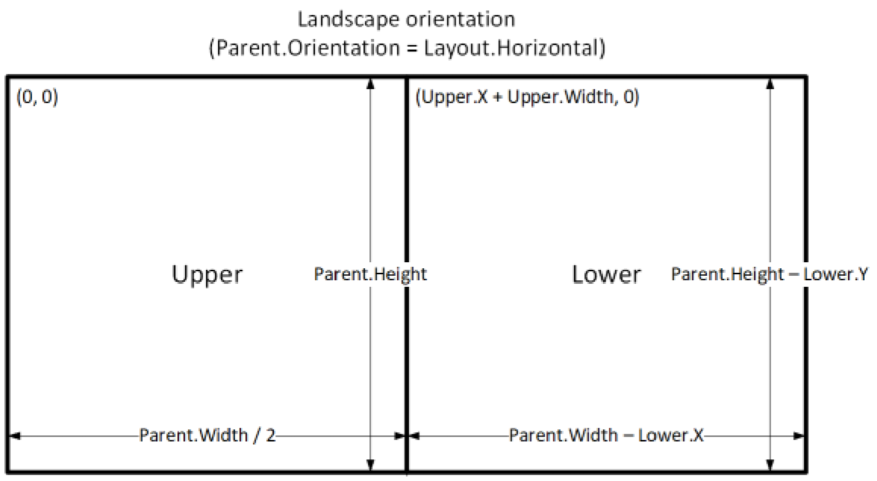
So far, you've learned how to use formulas to change each control's size in response to the available space, while keeping controls aligned relative to each other. But you might want or need to make more substantial layout changes in response to different device sizes and orientations. When a device is rotated from portrait to landscape orientation, for example, you might want to switch from a vertical layout to a horizontal one. On a larger device, you can present more content or rearrange it to provide a more appealing layout. On a smaller device, you might need to split up content across multiple screens.

Device orientation

You can use the screen's Orientation property to determine whether the screen is oriented vertically or horizontally.

Control	Property	Formula
Upper	**X**	`0`
Upper	**Y**	`0`
Upper	**Width**	`If(Parent.Orientation = Layout.Vertical, Parent.Width, Parent.Width / 2)`
Upper	**Height**	`If(Parent.Orientation = Layout.Vertical, Parent.Height / 2, Parent.Height)`
Lower	**X**	`If(Parent.Orientation = Layout.Vertical, 0, Upper.X + Upper.Width)`
Lower	**Y**	`If(Parent.Orientation = Layout.Vertical, Upper.Y + Upper.Height, 0)`
Lower	**Width**	`Parent.Width - Lower.X`
Lower	**Height**	`Parent.Height - Lower.Y`





Screen sizes and breakpoints

You can adjust your layout based on the size of the device. The screen's Size property classifies the current device size. The size is a positive integer; the ScreenSize type provides named constants to help with readability. This table lists the constants:

Constant	Value	Typical device type (using default app settings)
ScreenSize.Small	1	Phone
ScreenSize.Medium	2	Tablet, held vertically
ScreenSize.Large	3	Tablet, held horizontally
ScreenSize.ExtraLarge	4	Desktop computer

Use these sizes to make decisions about your app's layout. For example, if you want a control to be hidden on a phone-sized device but visible otherwise, you could set the control's **Visible** property to this formula:

```
Parent.Size >= ScreenSize.Medium
```

This formula evaluates to true when the size is medium or larger and false otherwise.

Performance

Apps built for Microsoft Teams are used slightly differently than standalone apps. Instead of persisting as a separate window, apps within Microsoft Teams are loaded just-in-time whenever a user switches between tabs. Because users are constantly switching through tabs as they complete their work, its particularly important to ensure that your apps are performant.

The following section reviews some of the key steps you can take to decrease the time it takes for your app to load. For a full list of optimizations, check out the doc article, Optimize canvas-app performance in Power Apps.

Limit data connections

Don’t connect to more than 30 data sources from the same app. Apps prompt new users to sign into each connector, so every additional connector increases the amount of time that the app needs to start. As an app runs, each connector requires CPU resources, memory, and network bandwidth when the app requests data from that source.

You can quickly measure your app’s performance by turning on Developer Tools in Microsoft Edge or Google Chrome while running the app. Your app is more likely to take longer than 15 seconds to return data if it frequently requests data from more than 30 data sources, such as Common Data Service, Azure SQL, SharePoint, and Excel on OneDrive.

Limit the number of controls

Don't add more than 500 controls to the same app. Power Apps generates an HTML DOM to render each control. The more controls you add, the more generation time Power Apps needs.

You can, in some cases, achieve the same result and have the app start faster if you use a gallery instead of individual controls. In addition, you might want to reduce the number of control types on the same screen. Some controls (such as PDF viewer, data table, and combo box) pull in large execution scripts and take longer to render.

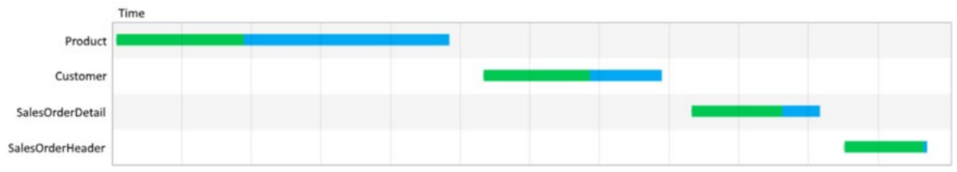
Optimize the OnStart function

You can use **Concurrent** function to cut the amount of time an app needs to load data in half.

Without the Concurrent function, this formula loads each of four tables one at a time:

```
ClearCollect( Product, '[Product]' );
ClearCollect( Customer, '[Customer]' );
ClearCollect( SalesOrderDetail, '[SalesOrderDetail]' );
ClearCollect( SalesOrderHeader, '[SalesOrderHeader]' )
```

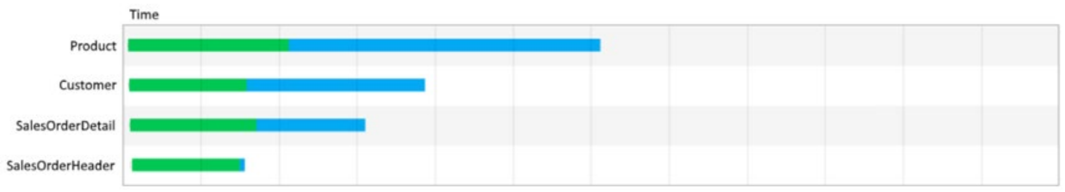
You can confirm this behavior in the Developer Tools for your browser:



You can enclose the same formula in the Concurrent function to reduce the overall time that the operation needs:

```
Concurrent(
      ClearCollect( Product, '[Product]' );
      ClearCollect( Customer, '[Customer]' );
      ClearCollect( SalesOrderDetail, '[SalesOrderDetail]' );
      ClearCollect( SalesOrderHeader, '[SalesOrderHeader]' )
)
```

With this change, the app fetches the tables in parallel:



Avoid controls dependency between screens

To improve performance, the screens of an app are loaded into memory only as they are needed. This optimization can be hampered if, for example, screen 1 is loaded and one of its formulas uses a property of a control from screen 2. Now screen 2 must be loaded to fulfill the dependency before screen 1 can be displayed. Imagine screen 2 has a dependency on screen 3, which has another dependency on screen 4, and so on. This dependency chain can cause many screens to be loaded.

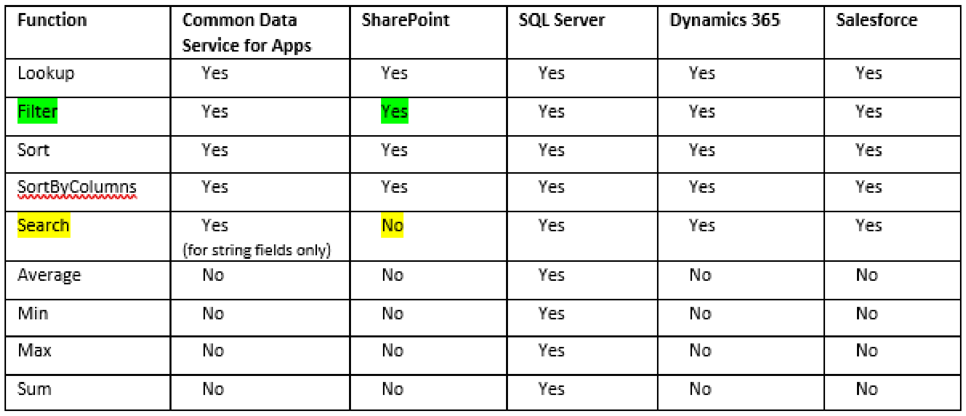
For this reason, avoid formula dependencies between screens. In some cases, you can use a global variable or collection to share information between screens.

There is an exception. In the previous example imagine that the only way to display screen 1 is by navigating from screen 2. Then screen 2 would have already been loaded in memory when screen 1 was to be loaded. No additional work is needed to fulfill the dependency for screen 2 and therefore there's no performance impact.

Use delegation

Where possible, use functions that delegate data processing to the data source instead of retrieving data to the local device for processing. If an app must process data locally, the operation requires much more processing power, memory, and network bandwidth, especially if the data set is large.

Different data sources support delegation from different functions:



Function	Common Data Service for Apps	SharePoint	SQL Server	Dynamics 365	Salesforce
Lookup	Yes	Yes	Yes	Yes	Yes
Filter	Yes	Yes	Yes	Yes	Yes
Sort	Yes	Yes	Yes	Yes	Yes
SortByColumns	Yes	Yes	Yes	Yes	Yes
Search	Yes (for string fields only)	No	Yes	Yes	Yes
Average	No	No	Yes	No	No
Min	No	No	Yes	No	No
Max	No	No	Yes	No	No
Sum	No	No	Yes	No	No

For example, SharePoint lists support delegation from the **Filter** function but not the **Search** function, so you should use **Filter** instead of **Search** to find items in a gallery if the SharePoint list contains more than 500 items.

Use Delayed Load

Turn on the experimental feature for **Delayed Load** if your app has more than 10 screens, no rules, and many controls that are on multiple screens and that are directly bound to the data source. If you build this type of app and don’t enable this feature, app performance may suffer because the controls in all screens must be populated even on screens that aren’t open. Also, all screens of the app must be updated whenever the data source changes, such as when the user adds a record.

Republish apps regularly

Republish your apps to get performance improvements and additional features from the Power Apps platform.

Avoid repeating the same formula in multiple places

If multiple properties run the same formula (especially if it's complex), consider setting it once and then referencing the output of the first property in subsequent ones. For example, don't set the **DisplayMode** property of controls A, B, C, D and E to the same complex formula. Instead, set A's **DisplayMode** property to the complex formula, set B's **DisplayMode** property to the result of A's **DisplayMode** property, and so on for C, D, and E.

Advanced scenarios

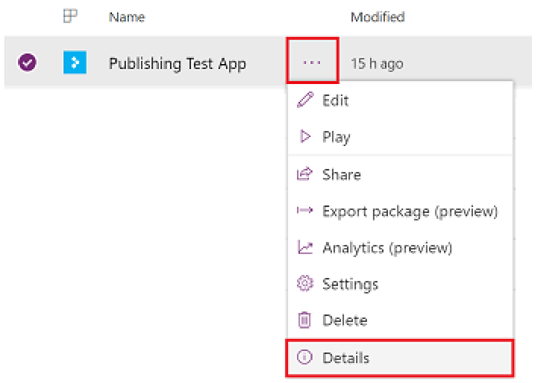
Creating a manifest from scratch

Sometimes it is necessary to build your own app manifest from Microsoft Teams instead of relying on the **Add to Teams** button within Power Apps. This can happen if you want to combine a Power Apps app with a bot or with other apps within a single app manifest file for Microsoft Teams.

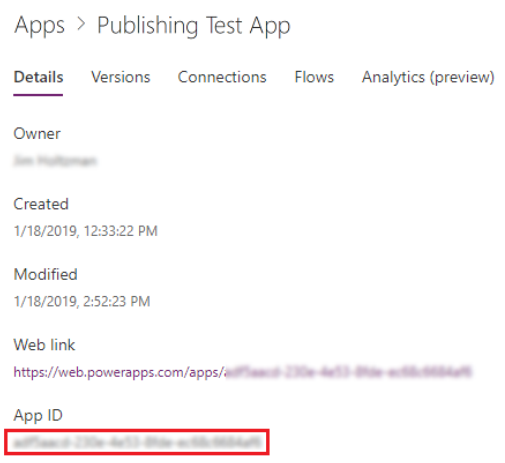
Locate your Power Apps app's GUID

To begin, you will want to find and make note of your Power Apps app's GUID for use in later steps. If you are creating a custom manifest for a model-driven app or a portal, you can skip this step.

1. Sign in to make.powerapps.com, and then select **Apps** in the menu.
2. Select **More Commands (...)** for the app you want to share in Microsoft Teams, and then select Details.



3. Record the **App ID** for later use.



Apps > Publishing Test App

Details Versions Connections Flows Analytics (preview)

Owner
[redacted]

Created
1/18/2019, 12:33:22 PM

Modified
1/18/2019, 2:52:23 PM

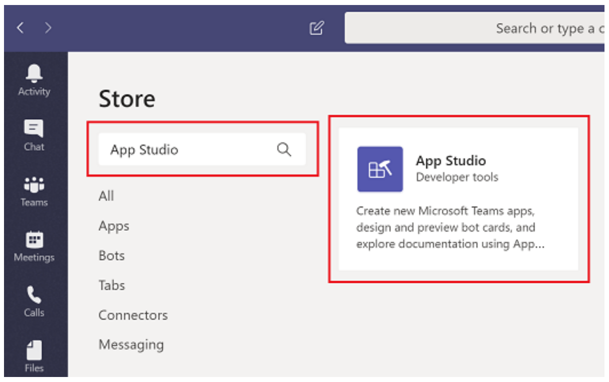
Web link
https://web.powerapps.com/apps/[redacted]

App ID
[redacted]

Install App Studio

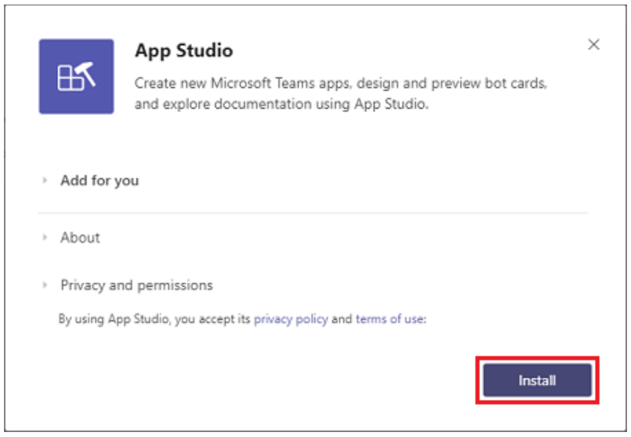
You can skip these steps if App Studio is already installed.

1. In Microsoft Teams, select **Apps** in the lower-left of the Microsoft Teams menu.
2. Search for "App Studio" in the search box and then select it.



Create new Microsoft Teams apps, design and preview bot cards, and explore documentation using App...

3. Select **Install**.



App Studio

Create new Microsoft Teams apps, design and preview bot cards, and explore documentation using App Studio.

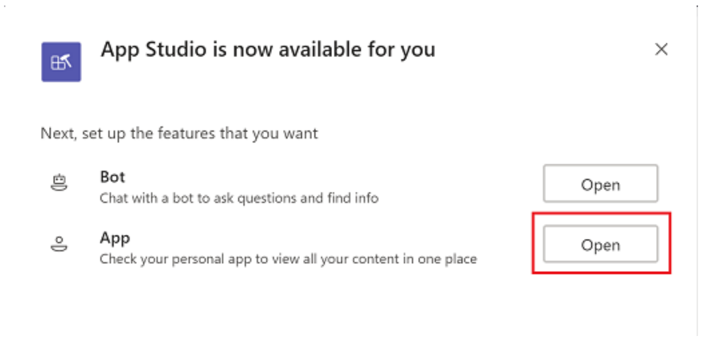
Add for you

About

Privacy and permissions

By using App Studio, you accept its privacy policy and terms of use:

4. Select **Open** for the App feature.



App Studio is now available for you

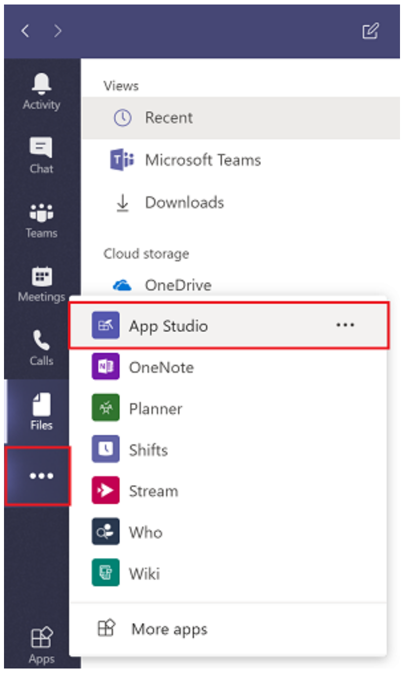
Next, set up the features that you want

Bot
Chat with a bot to ask questions and find info

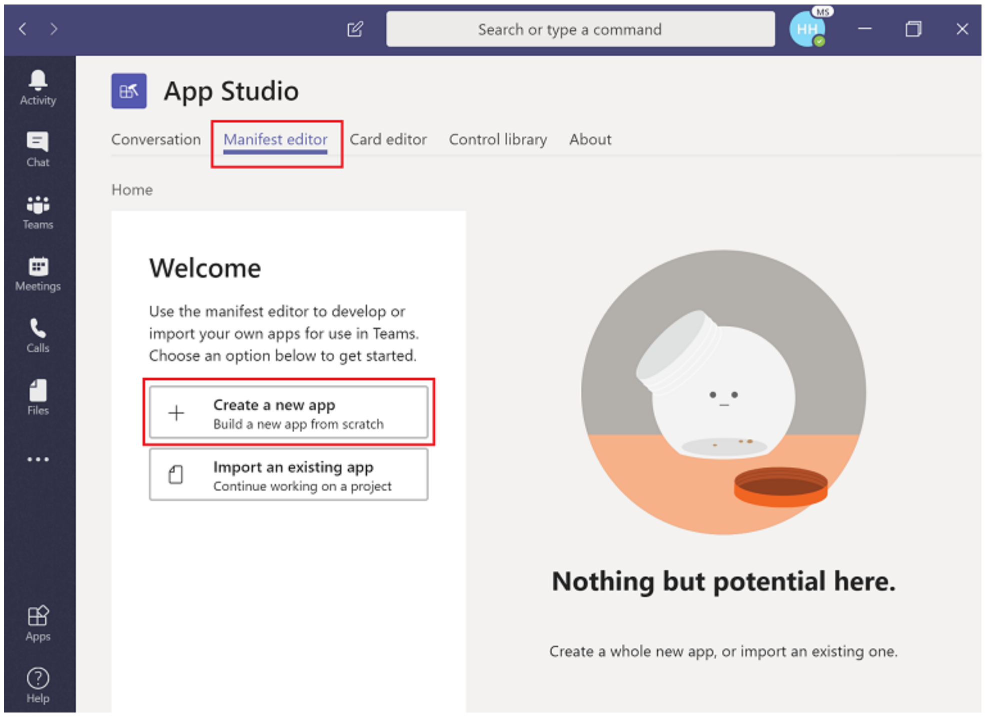
App
Check your personal app to view all your content in one place

Create a Microsoft Teams app for your Power Apps app

1. In Microsoft Teams, open **App Studio**.



2. Select the **Manifest editor** tab, and then select **Create** a new app under **Welcome**.



Welcome

Use the manifest editor to develop or import your own apps for use in Teams. Choose an option below to get started.

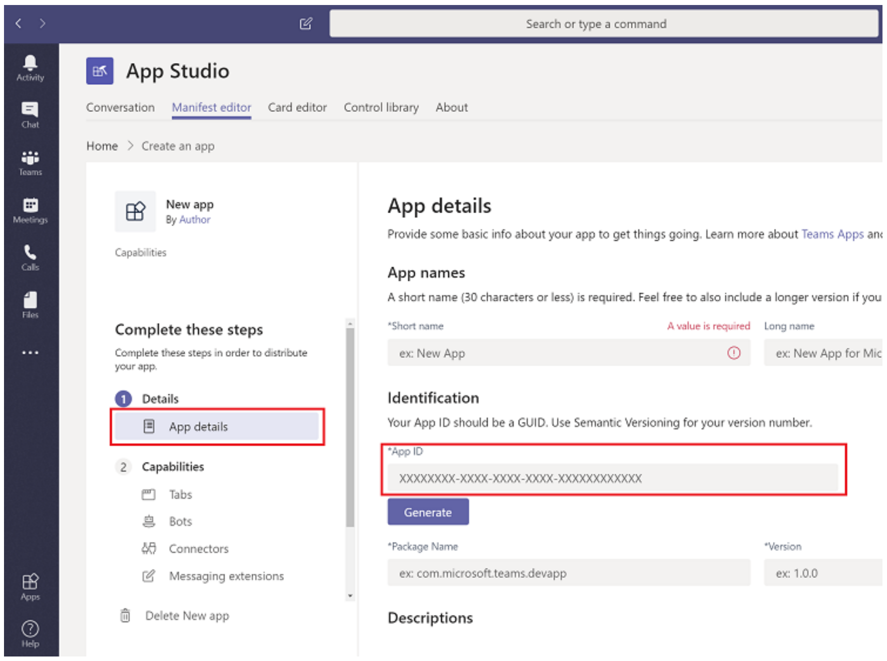
Create a new app
Build a new app from scratch

Import an existing app
Continue working on a project

Nothing but potential here.

Create a whole new app, or import an existing one.

3. Fill in information about your app in the **App details** page. For the App ID GUID, you should use your Power Apps **App ID** GUID you recorded above. This will avoid duplication of Microsoft Teams apps for a Power Apps app. If you are completing this form for a model-driven app or a portal, you can provide your own ID.



App Studio

Conversation | Manifest editor | Card editor | Control library | About

Home > Create an app

New app
By Author
Capabilities

Complete these steps
Complete these steps in order to distribute your app.

1 Details
App details

2 Capabilities
Tabs
Bots
Connectors
Messaging extensions

Delete New app

App details
Provide some basic info about your app to get things going. Learn more about Teams Apps an

App names
A short name (30 characters or less) is required. Feel free to also include a longer version if you

*Short name: ex: New App — A value is required
Long name: ex: New App for Mic

Identification
Your App ID should be a GUID. Use Semantic Versioning for your version number.

*App ID: XXXXXXXX-XXXX-XXXX-XXXX-XXXXXXXXXXXX
Generate

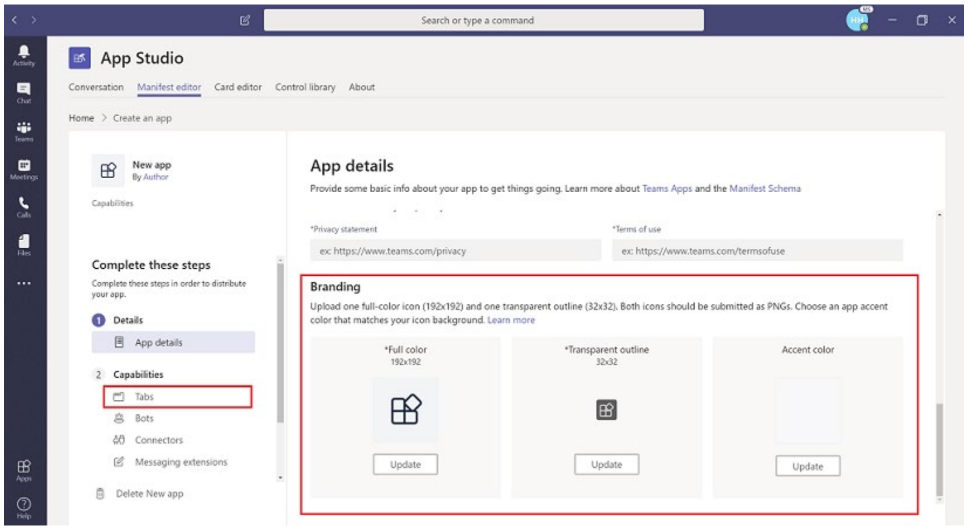
*Package Name: ex: com.microsoft.teams.devapp
*Version: ex: 1.0.0

Descriptions

Fields	Description
App names	
Short name	Required. The short display name for the app. 30 character limit.
Long name	The full name of the app, used if the full app name exceeds 30 characters.
Identification	
App ID	Required. The unique Microsoft-generated identifier for this app.
Package Name	Required. A unique identifier for this app. We recommend using reverse domain notation; for example, com.example.
Version	Required. The version of the specific app. If you update something in your manifest, the version must be incremented as well.
Descriptions	
Short description	Required. A short description of your app experience, used when space is limited. 80 character limit.
Long description	Required. The full description of your app.

Developer information	
Name	Required. The display name for the company or developer.
Website	Required. The https:// URL to the website for your app via Power Apps.com. When someone installs your app, an “About your app” page will appear. It should link to the web version of your app on Power Apps.com.
App URLs	These links will show up in the About page along with the website URL.
Privacy statement	Required. The https:// URL to the developer's privacy policy.
Terms of use	Required. The https:// URL to the developer's terms of use.
Branding	
Full color	A relative file path to a full color 192x192 PNG icon.
Transparent outline	A relative file path to a transparent 32x32 PNG outline icon.
Accent color	A color to use in conjunction with and as a background for your outline icons.

4. Scroll down to the **Branding** section and add your logos and the accent color desired for your app. These are the logos that will appear for your app in Microsoft Teams.



App details

Provide some basic info about your app to get things going. Learn more about Teams Apps and the Manifest Schema

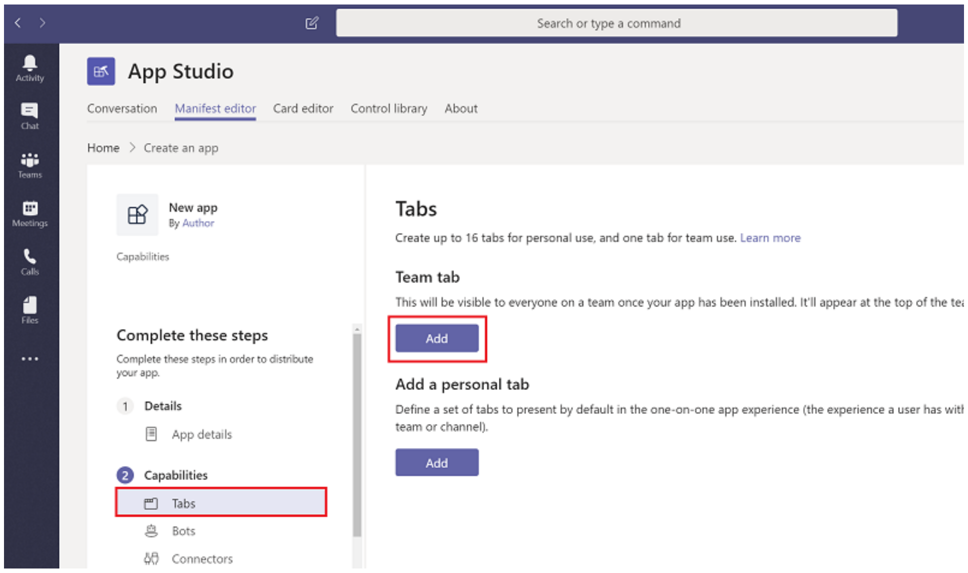
Complete these steps

Complete these steps in order to distribute your app.

Branding

Upload one full-color icon (192x192) and one transparent outline (32x32). Both icons should be submitted as PNGs. Choose an app accent color that matches your icon background. Learn more

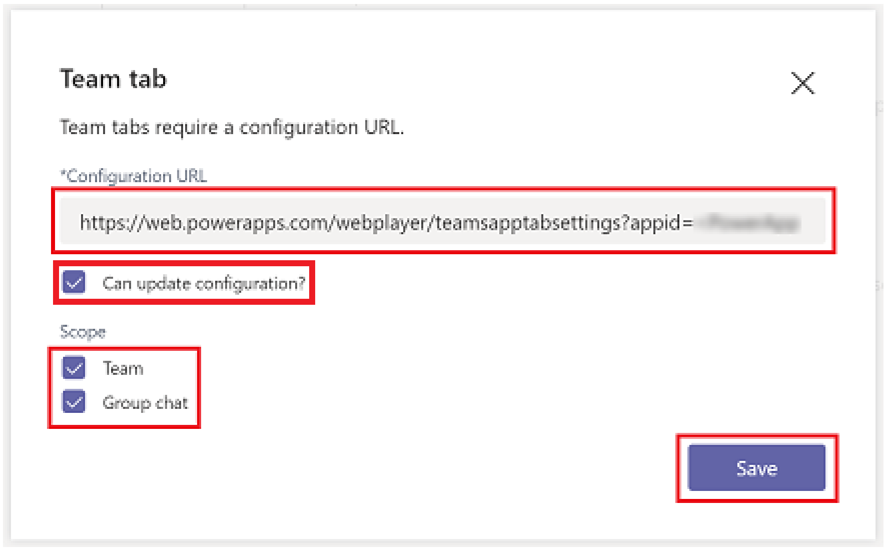
5. Under Capabilities, select **Tabs**.
6. Depending on which Microsoft Teams app type you want, either select **Add** under **Teams tab** or **Add a personal tab**.


Search or type a command
Activity
Chat
Teams
Meetings
Calls
Files
App Studio
Conversation
Manifest editor
Card editor
Control library
About
Home > Create an app
New app
By Author
Capabilities
Complete these steps
Complete these steps in order to distribute your app.
1 Details
App details
2 Capabilities
Tabs
Bots
Connectors
Tabs
Create up to 16 tabs for personal use, and one tab for team use. Learn more
Team tab
This will be visible to everyone on a team once your app has been installed. It'll appear at the top of the tea
Add
Add a personal tab
Define a set of tabs to present by default in the one-on-one app experience (the experience a user has wit team or channel).
Add

7. Add your app's configuration URL in the "Configuration URL" input field, using the following format: https://apps.powerapps.com/teams/settings/<PowerApp ID>

   If you are configuring a model-driven app or a portal, simply copy and paste the URL for your app.

8. Replace <Power App ID> with the **App ID** GUID you recorded above.
9. Select the scope for your app to appear in. Ensure Can update configuration is checked, and then select Save.



Team tab

Team tabs require a configuration URL.

*Configuration URL

https://web.powerapps.com/webplayer/teamsapptabsettings?appid=

Can update configuration? ✓

Scope

✓ Team

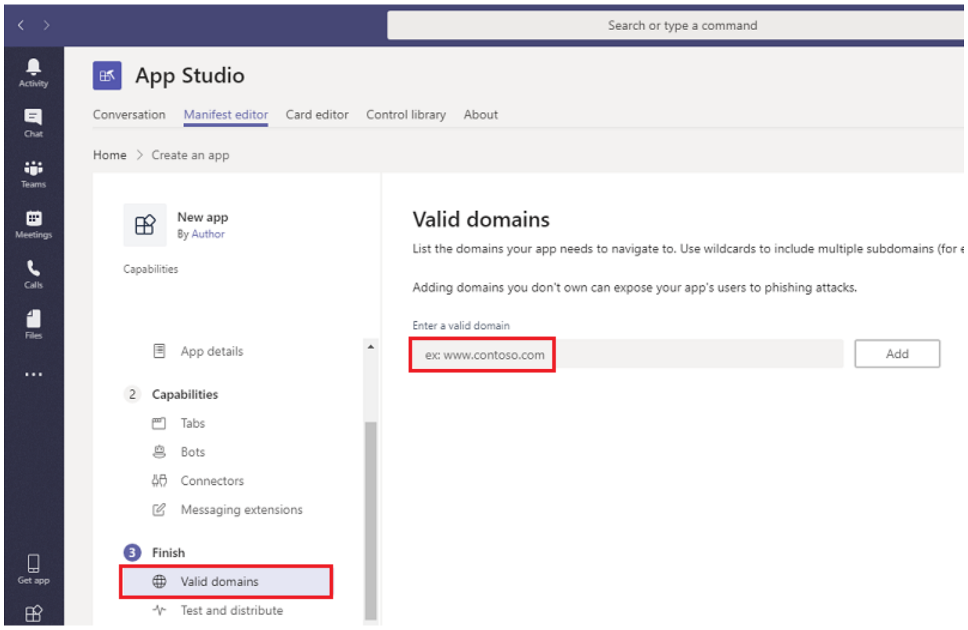
✓ Group chat

Save

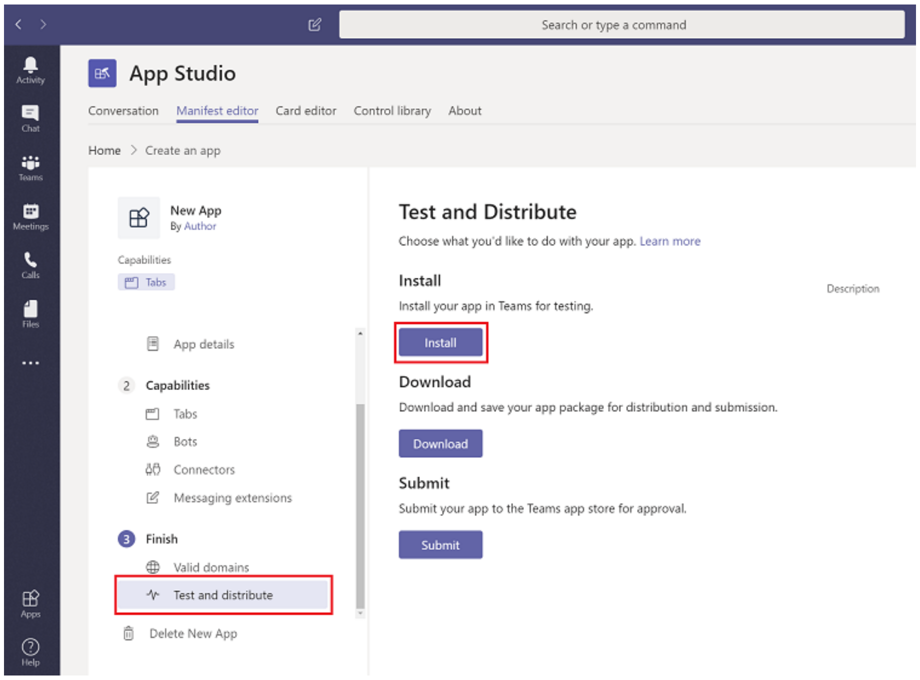
10. Under **Finish**, select **Valid domains**. Add apps.powerapps.com and apps.preview.powerapps.com as valid domains for the Microsoft Teams application.

If you are configuring a model-driven app, provide the domain of your CDS instance. For example, http://example.crm.dynamics.com.

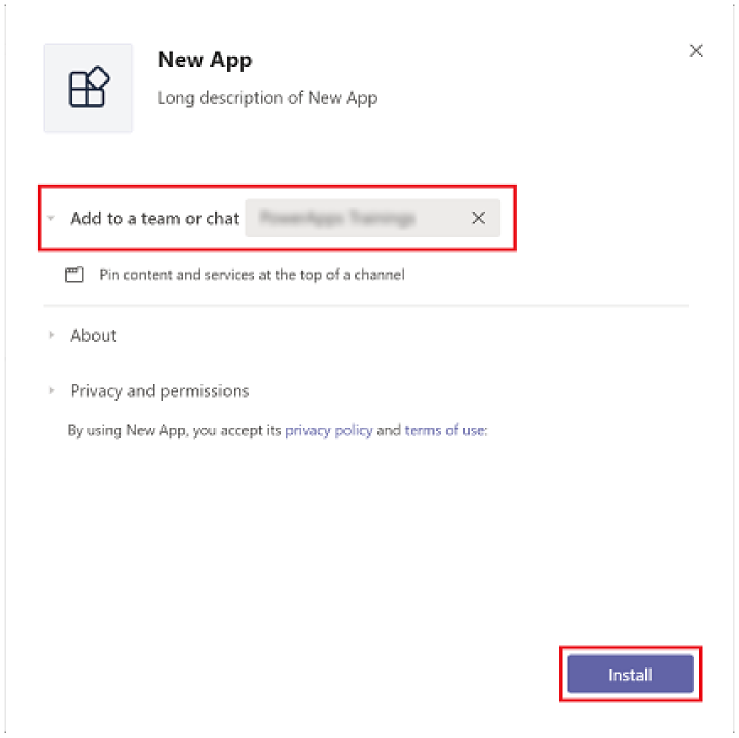
Similarly, if you are configuring a Power Apps portal, add the domain of your portal.



11. Under **Finish**, select **Test and distribute.** Under **Install**, select **Install**.

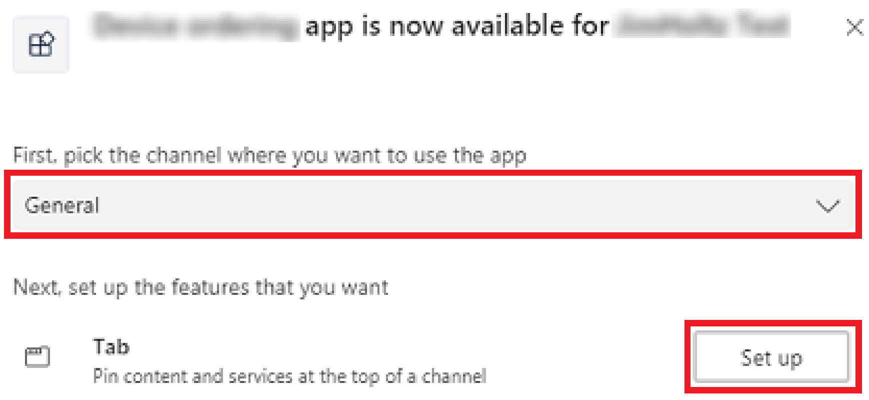


12. Select the team you want the app installed in, and then select **Install**.


New App
Long description of New App
Add to a team or chat
Pin content and services at the top of a channel
About
Privacy and permissions
Install

By using New App, you accept its privacy policy and terms of use.

13. If you want to add an instance of that app to a channel right away, select the channel you wish to use the app in and select Set up.



app is now available for

First, pick the channel where you want to use the app

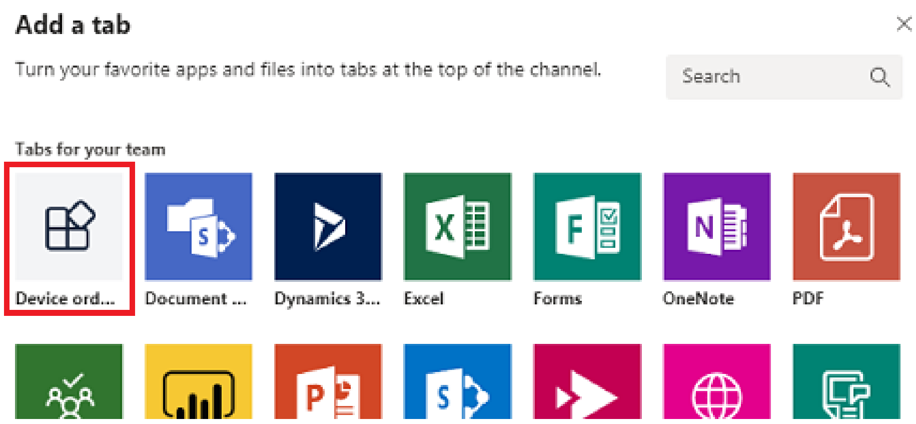
General

Next, set up the features that you want

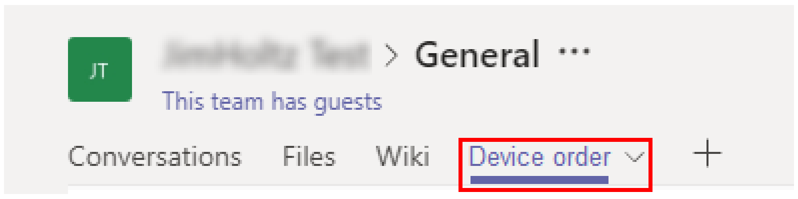
Tab
Pin content and services at the top of a channel

14. Select **Save**.
15. Add the app as a tab

16. To add the app as a tab to any channel or conversation, select **+**, and then under **Tabs for your team** select your app.



17. The app now appears as a tab.


JT
General
This team has guests
Conversations
Files
Wiki
Device order