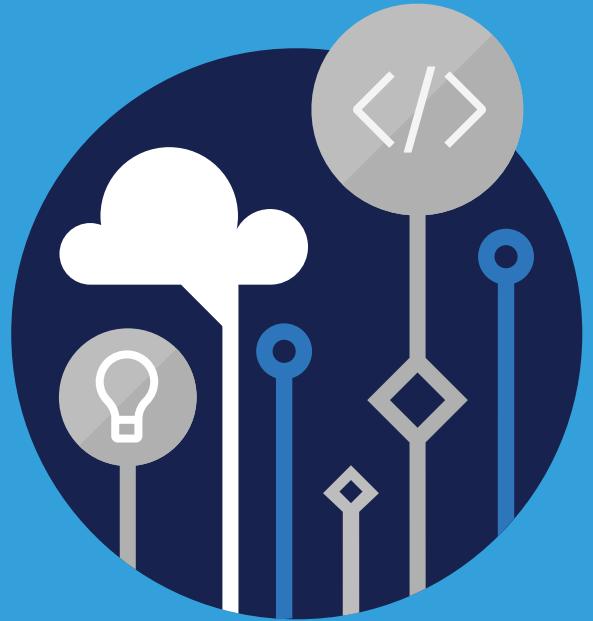


Microsoft
Official
Course



PL-100T00

Microsoft Power Platform
App Maker

PL-100T00

**Microsoft Power Platform App
Maker**

II Disclaimer

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2019 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/trademarks>¹ are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

¹ <http://www.microsoft.com/trademarks>

MICROSOFT LICENSE TERMS

MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which includes the media on which you received it, if any. These license terms also apply to Trainer Content and any updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms apply.

**BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS.
IF YOU DO NOT ACCEPT THEM, DO NOT ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.**

If you comply with these license terms, you have the rights below for each license you acquire.

1. DEFINITIONS.

1. "Authorized Learning Center" means a Microsoft Imagine Academy (MSIA) Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.
2. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.
3. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns or controls that is located at an Authorized Learning Center's training facilities that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
4. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session or Private Training Session, (ii) an employee of an MPN Member (defined below), or (iii) a Microsoft full-time employee, a Microsoft Imagine Academy (MSIA) Program Member, or a Microsoft Learn for Educators – Validated Educator.
5. "Licensed Content" means the content accompanying this agreement which may include the Microsoft Instructor-Led Courseware or Trainer Content.
6. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program.
7. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics, or Microsoft Business Group courseware.
8. "Microsoft Imagine Academy (MSIA) Program Member" means an active member of the Microsoft Imagine Academy Program.
9. "Microsoft Learn for Educators – Validated Educator" means an educator who has been validated through the Microsoft Learn for Educators program as an active educator at a college, university, community college, polytechnic or K-12 institution.
10. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.
11. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals, developers, students at an academic institution, and other learners on Microsoft technologies.
12. "MPN Member" means an active Microsoft Partner Network program member in good standing.

13. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.
 14. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.
 15. "Trainer" means (i) an academically accredited educator engaged by a Microsoft Imagine Academy Program Member to teach an Authorized Training Session, (ii) an academically accredited educator validated as a Microsoft Learn for Educators – Validated Educator, and/or (iii) a MCT.
 16. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.
2. **USE RIGHTS.** The Licensed Content is licensed, not sold. The Licensed Content is licensed on a **one copy per user basis**, such that you must acquire a license for each individual that accesses or uses the Licensed Content.
- 2.1 Below are five separate sets of use rights. Only one set of rights apply to you.
 1. **If you are a Microsoft Imagine Academy (MSIA) Program Member:**
 1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
 2. For each license you acquire on behalf of an End User or Trainer, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
 2. provide one (1) End User with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content.
 3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
 3. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End

User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Authorized Training Sessions,
6. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and
7. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

2. If you are a Microsoft Learning Competency Member:

1. Each license acquire may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or MCT, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Authorized Training Session and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**
 2. provide one (1) End User attending the Authorized Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. you will provide one (1) MCT with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure that each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
 3. you will ensure that each End User provided with a hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

4. you will ensure that each MCT teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
5. you will only use qualified MCTs who also hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Authorized Training Sessions using MOC,
6. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
7. you will only provide access to the Trainer Content to MCTs.

3. If you are a MPN Member:

1. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
2. For each license you acquire on behalf of an End User or Trainer, you may either:
 1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
 2. provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
 3. you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content.
3. For each license you acquire, you must comply with the following:
 1. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
 2. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,
 3. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
 4. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,

5. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,
6. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,
7. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
8. you will only provide access to the Trainer Content to Trainers.

4. If you are an End User:

For each license you acquire, you may use the Microsoft Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led Courseware is in digital format, you may access the Microsoft Instructor-Led Courseware online using the unique redemption code provided to you by the training provider and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

5. If you are a Trainer.

1. For each license you acquire, you may install and use one (1) copy of the Trainer Content in the form provided to you on one (1) Personal Device solely to prepare and deliver an Authorized Training Session or Private Training Session, and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Trainer Content. You may not install or use a copy of the Trainer Content on a device you do not own or control. You may also print one (1) copy of the Trainer Content solely to prepare for and deliver an Authorized Training Session or Private Training Session.

2. If you are an MCT, you may customize the written portions of the Trainer Content that are logically associated with instruction of a training session in accordance with the most recent version of the MCT agreement.
3. If you elect to exercise the foregoing rights, you agree to comply with the following: (i) customizations may only be used for teaching Authorized Training Sessions and Private Training Sessions, and (ii) all customizations will comply with this agreement. For clarity, any use of "customize" refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

- 2.2 **Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.
- 2.3 **Redistribution of Licensed Content.** Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.
- 2.4 **Third Party Notices.** The Licensed Content may include third party code that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code are included for your information only.
- 2.5 **Additional Terms.** Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.

3. **LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY.** If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("Pre-release"), then in addition to the other provisions in this agreement, these terms also apply:
 1. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.
 2. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its technology, technologies, or products to third parties because we include your feedback in them. These rights survive this agreement.
 3. **Pre-release Term.** If you are an Microsoft Imagine Academy Program Member, Microsoft Learning Competency Member, MPN Member, Microsoft Learn for Educators – Validated Educator, or Trainer, you will cease using all copies of the Licensed Content on the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using the Licensed Content on the Pre-release technology, or (ii) sixty (60) days after the commercial release of the technology that is the subject of the Licensed Content, whichever is earliest ("Pre-release term"). Upon expiration or termination of the Pre-release term, you will irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.
4. **SCOPE OF LICENSE.** The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
 - access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
 - alter, remove or obscure any copyright or other protective notices (including watermarks), branding or identifications contained in the Licensed Content,
 - modify or create a derivative work of any Licensed Content,
 - publicly display, or make the Licensed Content available for others to access or use,
 - copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
 - work around any technical limitations in the Licensed Content, or
 - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation.
5. **RESERVATION OF RIGHTS AND OWNERSHIP.** Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property

laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.

6. **EXPORT RESTRICTIONS.** The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.
7. **SUPPORT SERVICES.** Because the Licensed Content is provided "as is", we are not obligated to provide support services for it.
8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.
9. **LINKS TO THIRD PARTY SITES.** You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.
10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.
11. **APPLICABLE LAW.**
 1. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.
 2. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.
12. **LEGAL EFFECT.** This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.
13. **DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.**
14. **LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US\$5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.**

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential, or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 \$ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.

Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et.
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised April 2019



Contents

■	Module 0 Welcome	1
	Start Here	1
■	Module 1 Introduction to Power Platform	3
	Power Platform overview	3
	Module summary	8
■	Module 2 Create a model-driven app in Power Apps	11
	Introduction to model-driven apps and Dataverse	11
	Get started with Dataverse	17
	Get started with model-driven apps in Power Apps	31
	Create and manage columns within a table in Dataverse	49
	Get started with security roles in Dataverse	61
	Document and test your Power Apps application	73
	Get started with Microsoft Dataverse for Teams	85
	Build your first app with Power Apps for Teams	99
■	Module 3 Create a canvas app in Power Apps	115
	Get started with Power Apps	115
	Customize a canvas app in Power Apps	142
	Manage apps in Power Apps	153
	Navigation in a canvas app in Power Apps	160
	How to build the UI in a canvas app in Power Apps	166
	Use and understand Controls in a canvas app in Power Apps	175
■	Module 4 Automate a business process using Power Automate	185
	Get started with Power Automate	185
	Use the Admin center to manage environments and data policies in Power Automate	214
	Introduction to business process flows in Power Automate	230
	Create an immersive business process flow in Power Automate	234
	Introduction to Power Automate security and governance	255
	Build your first workflow with Power Automate for Teams	268
■	Module 5 Create and use analytics reports with Power BI	287
	Get started with Power BI	287
	Get data with Power BI Desktop	302
	Create reports with Power BI for Teams	320

■	Module 6 AI Builder and Power Virtual Agents	355
	Get started with AI Builder	355
	Create a chatbot with Power Virtual Agents for Teams	365

Module 0 Welcome

Start Here

About this course

Course Description

This course will teach you how to build apps with low-code techniques to simplify, automate, and transform business tasks and processes using Microsoft Power Platform.

Audience Profile

The App Maker builds solutions to simplify, automate, and transform tasks and processes for themselves and their team where they have deep expertise in the solution business domain. They have basic data modeling, user experience design, requirements analysis, and process analysis skills.

The App Maker creates and enforces business processes, structures digital collection of information, improves efficiency of repeatable tasks, and automates business processes.

The App Maker uses the Maker tools of Power Platform to solve business problems. They may use advanced features of Microsoft apps and third-party productivity tools.

The App Maker is aware of the capabilities and limitations of available tools and understands how to apply them. The App Maker is self-directed, and solution focused. They may not have formal IT training but are comfortable using technology to solve business problems with a personal growth mindset. They understand the operational need and have a vision of the desired outcome. They approach problems with phased and iterative strategies.

Course Completion

After completing this course, students will be able to:

- Design apps and automate workflows
- Create apps and automate workflows

- Analyze and visualize data in context of an app or automated workflow
- Implement and manage apps and automated workflows

Power Platform App Maker

Certification

Upon completion of this course and practicing building solutions, we encourage you to [get certified.¹](#)

PL-100 Certification Exam

This exam measures your ability to accomplish the following technical tasks: design solutions; create solutions; analyze data; and implement and manage solutions.

PL-100 Study Areas	Weights
Design solutions	35-40%
Create solutions	45-50%
Analyze and visualize data	15-20%

PL-100 Power Platform App Maker lab

This course is accompanied by a business case lab that can be accessed at <https://github.com/MicrosoftLearning/PL-100-Microsoft-Power-Platform-App-Maker>

¹ <https://docs.microsoft.com/en-us/learn/certifications/power-platform-app-maker>

Module 1 Introduction to Power Platform

Power Platform overview

Introduction to the Power Platform

Modern businesses run on data. Users interact with data daily from entering their time for payroll, seeking guidance on existing processes, and analyzing data to make decisions. In our technology driven world, users can be empowered to gain insights from and interact with data all while automating those menial responsibilities that seem to be more burden than job task. Power Platform enables your business to craft solutions while empowering you to unite customized technology to help everyone, from the CEO to the front-line workers, drive the business with data.

In this lesson, you will:

- Learn the components and features of Power Platform
- Identify when to use each Power Platform component application to create business solutions
- Learn the value of using the Power Platform to create business solutions

What is the Power Platform

Power Platform is comprised of four key products: Power Apps, Power Automate, and Power BI and Power Virtual Agents.

Power Apps

Power Apps provides a rapid low code development environment for building custom apps for business needs. It has services, connectors, and a scalable data service and app platform (Dataverse) to allow simple integration and interaction with existing data. Power Apps enables the creation of web and mobile applications that run on all devices.

People use apps for every area of their lives, and business should be no exception. Most out of the box solutions do not meet exact business needs or integrate well with other business programs. Power Apps eases users into app development with a simple interface so that every business user or pro developer can build custom apps.

Power Automate

Power Automate lets users create automated workflows between applications and services. It helps automate repetitive business processes such as communication, data collections, and decision approvals. Don't waste important productive hours on drafting the same email for a weekly update or walking approvals through. Not only for the individual user, Power Automate allows for the creation of enterprise-grade process automation. Power Automate's simple interface allows every level of user to automate work tasks from beginners to seasoned developers.

Power BI

Power BI (Business Intelligence) is a business analytics service that delivers insights for analyzing data. It can share those insights through data visualizations which make up reports and dashboards to enable fast, informed decisions. Power BI scales across an organization, and it has built-in governance and security allowing businesses to focus on using data more than managing it. You can consider Power BI as the analysis and insights leg of the Power Platform. It takes business data and allows you to display it in ways that makes the most sense to users. A Power BI dashboard could potentially replace a standing meeting to report out on company metrics such as sales data, progress against goals, or employee performance.

Power Virtual Agents

Power Virtual Agents enables anyone to create powerful chatbots using a guided, no-code graphical interface, without the need for data scientists or developers.

Power virtual agents addresses many of the major issues with chatbot building. It eliminates the gap between subject matter experts and the development teams building the chatbots. It removes the complexity of exposing teams to the nuances of conversational AI and the need to write complex code. It minimizes the IT effort required to deploy and maintain a custom conversational solution by empowering subject matter experts to build and maintain their own conversational solutions.

Features

Among the programs listed above, there are cross cutting features which enable the Power Platform to be leveraged to its full potential. Some of these are:

AI Builder

AI Builder lets users and developers add AI capabilities to the workflows and PowerApps they create and use. AI Builder is a turnkey solution that allows you to easily add intelligence to your workflows and apps and predict outcomes to help improve business performance without writing code.

The Dataverse

Dataverse is a scalable data service and app platform which lets users securely store and manage data from multiple sources and integrate that data in business applications using a common data model to ensure ease and consistency to users. Dataverse is the common currency that enables the components of Power Platform to work together. It's the foundation that enables the consolidation, display, and manipulation of data.

Connectors

Connectors enable you to connect apps, data, and devices in the cloud. Consider connectors the bridge across which information and commands travel. There are more than 275 connectors for the Power Platform, enabling all of your data and actions to connect cohesively. Examples of popular connectors include Salesforce, Office 365, Twitter, Dropbox, Google services, and more.

Portals

Portals bring the power of no-code solutions to building externally facing websites. Through the Power Apps interface, you can build an anonymous or authenticated website that allows users to interact with data held in Dataverse. The same drag and drop experience you enjoy when building apps is available to build these rich, interactive websites.

Although every feature is essential to building powerful solutions, let's dive in deeper to one of the features of Power Platform, **connectors**.

Data connectors

Power Platform is made powerful by its ability to leverage data across many platforms. To do this, components of the Power Platform use connectors. You can think of connectors as a bridge from your data source to your app or workflow which allows information to be conveyed back and forth. Connectors allow you to extend your business solutions across platforms and add functionality for your users, and Power Platform has more than 275 connectors with the ability to build custom connectors as well.

It is important to note that premium connectors require a paid plan for use. The connector reference in the summary and resources unit lists all connectors and whether they are considered standard or premium. Custom connectors are also considered a premium feature.

Data Sources

In order to understand the types of connectors and what you can do with them, you must first understand the types of data sources to which they connect. The two types of data sources are tabular and function-based.

Tabular data - A tabular data source is one that returns data in a structured table format. Power Apps can directly read and display these tables through galleries, forms, and other controls. Additionally, if the data source supports it, Power Apps can create, edit, and delete data from these data sources. Examples include Dataverse, SharePoint, and SQL Server.

Function-based data - A function-based data source is one that uses functions to interact with the data source. These functions can be used to return a table of data, but offer more extensive action such as the ability to send an email, update permissions, or create a calendar event. Examples include Office 365 Users, Project Online, and Azure Blob Storage.

Both of these data source types are commonly used to bring data and additional functionality to your solutions.

As you can see, connecting to data sources allows you to integrate disparate parts of your business solutions to build them out cohesively.

Connectors

Now that you understand more about data sources and delegation, you are ready to learn about connectors.

Connectors are the bridges from your data source to your app, workflow, or dashboard. The Power Platform has more than 275 connectors available to common data sources. Connectors are divided into standard and premium. Some popular standard connectors are SharePoint, Outlook, and YouTube. Premium connectors require additional licensing for your app and/or users. A few premium connectors are SQL Server, Survey Monkey, and Mail Chimp.

Triggers and Actions

Once you have established a data source and configured your connector, there are two types of operations you can use, triggers or actions.

Triggers are only used in Power Automate and prompt a flow to begin. Triggers can be time based, such as a flow which begins every day at 8:00 am, or they could be based off of an action like creating a newrow in a table or receiving an email. You will always need a trigger to tell your workflow when to run.

Actions are used in Power Automate and Power Apps. Actions are prompted by the user or a trigger and allow interaction with your data source by some function. For example, an action would be sending an email in your workflow or app or writing a new line to a data source.

Now that you understand what connectors are and how to use them, let's look at what to do when there isn't a connector already built for your data source.

Custom Connectors

While the Power Platform offers more than 200 connectors, you also have the option to build a custom connector. This will allow you to extend your app by calling a publicly available API, or a custom API you're hosting in a cloud provider, such as Azure. API stands for Application Programming Interface and holds a series of functions available for developers. Connectors work by sending information back and forth across these APIs and gathering available functions into Power Apps or Power Automate. Because these connectors are function-based, they will call specific functions in the underlying service of the API to return the corresponding data.

An advantage of building custom connectors is that they can be used in different platforms, such as PowerApps, Power Automate, and Azure Logic Apps.

Creating Custom Connectors

You can create custom connectors using 3 different approaches:

- **Using a blank custom connector¹**
- **From an OpenAPI definition²**
- **From a Postman collection³**

While the requirements for each approach will vary, they all require a Power Apps per app or per user plan. Each link above points to the instructions for each approach.

¹ <https://docs.microsoft.com/connectors/custom-connectors/define-blank>

² <https://docs.microsoft.com/connectors/custom-connectors/define-openapi-definition>

³ <https://docs.microsoft.com/connectors/custom-connectors/define-postman-collection>

Note: The purpose of this module is to help you better understand data sources and connectors as a whole, but if you would like to learn more about custom connectors and even walk through an exercise to build one, check out the module Use custom connectors in a Power Apps canvas app.

Pulling it together

Although we live in a data driven world, your business can find it difficult to take advantage of the data you have access to. Sales, customer, and employee data should drive our business decisions, but where do we even start? The Power Platform can add value to any business by helping you to analyze, act, and automate. Act by building custom apps in Power Apps, automate processes based on the data you collect in Power Automation, and analyze the data you have collected in Power BI.

Consider a business that has IT equipment for general use. Currently equipment check-out is conducted by visiting the IT office and if the product is available, writing your name and the equipment name in a notebook. Employees may have to visit IT several times before equipment becomes available, and IT personnel must drop their tasks to check on equipment status or go to collect it for the employee. Sometimes employees hold onto the equipment longer than they intend and an IT personnel spends time tracking it down. In addition, important equipment information such as serial number, warranty details, and instructions for use are kept somewhere in the IT office. How can the Power Platform improve this process?

Power Apps allows us to build an app that has all equipment listed, the status of that equipment, and even important details such as use instructions. This way employees can check out available equipment, walk to IT at a specified pick up time where the equipment will be ready, and even access the use instructions or flag an equipment malfunction from their phone or tablet. Power Automate can read when equipment needs to be returned and send out reminder emails, or even a warning that the equipment is late being checked in. Users can see when equipment is booked through the app and request check out for a future date at which time Power Automate can send them a reminder to pick up the equipment and IT a reminder to have it ready. Power BI can take all the data generated from the app and analyze it to help you understand what equipment is used most often and by whom. This way you can decide if you need additional equipment, if some users or departments need dedicated equipment, and when your equipment has reached the end of its usefulness.

This is only one common scenario in which the Power Platform can transform the way businesses work. Consider your own business and what processes take up valuable time and are a burden to customers or employees. How can you leverage the Power Platform to improve them?

Module summary

Questions

Multiple choice

Your social media engagement officer has requested your help in boosting followers and retweets on Twitter. How could you help her get more information to better understand and subsequently increase engagement? <<

- Power Apps portals can create a new customer site for our followers
- Power Automate can handle our content approvals for us, reducing the time it takes to produce new content and ensure our quality procedure is followed
- Configure a Power BI report to capture and analyze data from Twitter, allowing you to better understand why certain posts elicit more responses

Multiple choice

Your team has become frustrated with number of times they have to perform basic data entry on project startup. There are many divisions who need the information and sometimes human error results in mistakes, making it more difficult to make sense of your information. Which program would be the most help in this situation? <<

- Power Apps
- Power Automate
- Power BI

Multiple choice

Someone has added an item in SharePoint which prompts a workflow to run in Power Automate. What type of operation have you used to start your workflow? <<

- Trigger
- Action
- Function-based

Multiple choice

A client likes the idea of implementing a Power Platform solution, but is concerned about the ability to interact with a custom API. How should you respond? <<

- The Power Platform offers the ability to create custom connectors for this purpose, which allow you to connect to Power Apps and Power Automate.
- The Power Platform has over 270 connectors to use in these situations.
- The Power Platform uses connectors that hold a series of functions available for developers.

Summary

Microsoft Power Platform offers a point-and-click approach to building custom applications, data visualizations, and automated workflows. This approach makes it easy for anyone familiar with Microsoft Office to create custom business solutions.

Now that you have reviewed this module, you should be able to:

- Describe the components and features of the Power Platform
- Identify when to use each Power Platform component application to create business solutions
- Understand and explain the value of using the Power Platform to create business solutions

Key takeaways

Here are the five key takeaways:

1. Power Platform is a system that enables users to do three key actions on data that help them drive business: gain insights from data (Analyze), drive intelligent business processes via apps they build (Act), and automate business processes (Automate).
2. Power BI helps you analyze and visualize data on a unified platform with data from internal and external sources.
3. Power Apps helps you build and deploy customized apps that work across web and mobile, embedded or standalone, on any device.
4. Connectors are bridges that allow you to send information from your data source to your app or workflow and back.
5. Power Automate helps you create automation workflows, from simple to advanced scenarios.

Answers

Multiple choice

Your social media engagement officer has requested your help in boosting followers and retweets on Twitter. How could you help her get more information to better understand and subsequently increase engagement? <<

- Power Apps portals can create a new customer site for our followers
- Power Automate can handle our content approvals for us, reducing the time it takes to produce new content and ensure our quality procedure is followed
- Configure a Power BI report to capture and analyze data from Twitter, allowing you to better understand why certain posts elicit more responses

Explanation

Power BI allows you to create visuals and better understand your data. Once you understand trends in what followers like, you can post more of that content and increase engagement.

Multiple choice

Your team has become frustrated with number of times they have to perform basic data entry on project startup. There are many divisions who need the information and sometimes human error results in mistakes, making it more difficult to make sense of your information. Which program would be the most help in this situation? <<

- Power Apps
- Power Automate
- Power BI

Explanation

Power Automate can create automated information workflows so that data entry only has to occur once.

Multiple choice

Someone has added an item in SharePoint which prompts a workflow to run in Power Automate. What type of operation have you used to start your workflow? <<

- Trigger
- Action
- Function-based

Explanation

A trigger is an operation that tells a workflow to begin or prompts some type of action.

Multiple choice

A client likes the idea of implementing a Power Platform solution, but is concerned about the ability to interact with a custom API. How should you respond? <<

- The Power Platform offers the ability to create custom connectors for this purpose, which allow you to connect to Power Apps and Power Automate.
- The Power Platform has over 270 connectors to use in these situations.
- The Power Platform uses connectors that hold a series of functions available for developers.

Explanation

You can build out a custom connector to bridge your app or workflow to the API.

Module 2 Create a model-driven app in Power Apps

Introduction to model-driven apps and Dateverse

Introduction to Dateverse

Whether you are building a canvas app or a model-driven app, Dataverse is the ideal data source because it is

the foundational data source of Microsoft Power Platform. As a result, you will experience the most functionality, the deepest integrations, the most features, and the best ease-of-use of any available data sources. From simple web-based data design to robust, role-based security,

Dataverse is a straightforward platform that you can use to begin designing your data structures and helping to keep them safe. Then, with your data in place, you have rich integration capabilities from Power Apps and the rest of

Microsoft Power Platform. Additionally, by applying business rules, you can trust that your business integrity will be maintained no matter what tool you use to interact with the data.

Another benefit of using Dataverse is that all of your data is stored in tables. A table is a set of records that is used to store data, similar to how a table stores data within a database.

Dataverse includes a base set of standard tables that cover typical scenarios, but you can also create custom tables that are specific to your organization. Standard and custom tables within Dataverse help provide a secure and cloud-based storage option for your data.

tables allow you to create a business-focused definition of your organization's data for use within apps.

Some benefits of using Dataverse and its tables include:

- **Simple to manage** - Both the metadata and data are stored in the cloud so that you're confident about the details of how they're stored.
- **Helps secure your data** - Data is stored so that users can see it only if you grant them access. Role-based security allows you to control access to tables for different users within your organization.
- **Access your Dynamics 365 data** - If you use Dynamics 365, data from your Dynamics 365 application is also stored within Dataverse, allowing you to quickly build apps that use your Dynamics 365 data and extend your apps by using Power Apps.
- **Rich metadata** - Data types and relationships are used directly within Power Apps.
- **Logic and validation** - Define calculated columns, business rules, workflows, and business process flows to ensure data quality and drive business processes.
- **Productivity tools** - tables are available within the add-ins for Microsoft Excel to increase productivity and ensure data accessibility.

Now that you have a better understanding of how Dataverse works and are aware of some of the benefits, you can now explore how these benefits can be applied in a model-driven app.

Introduction to model-driven apps

Unlike canvas apps, where you build out an app screen-by-screen by adding logic and code as you go, model-driven apps can be created with a few simple steps. Model-driven apps use a component-focused approach to develop the app. When developing canvas apps, you have complete control over the appearance and behavior of your app, whereas with model-driven apps, the layout is mainly based on the components that you add to the app.

With model-driven apps, a number of different components and component properties are available for you to add and modify when designing an app.

Model-driven app design provides the following benefits:

- Rich component-focused, no-code design environments
- An ability to create complex responsive apps with a similar UI across a variety of devices from desktop to mobile
- Robust design capability
- Apps can be distributed as a solution

Dataverse and model-driven apps working together

When creating a model-driven app, you can use tables from Dataverse as your building blocks. Model-driven apps start with your data model, building up from the shape of your core business data and processes in Dataverse to model forms, views, and other components.

It's important to ensure that your business data and business processes at the data level are structured properly before you compose your app. Model-driven apps will automatically generate a UI that is responsive across devices; however, this outcome relies heavily on how your data is modeled in Dataverse.

Approach to model-driven app making

When creating model-driven apps, it's important to focus on three areas:

- Modeling business data
- Defining business processes
- Composing the app.

More information on creating model-driven apps is included in the Create relationships, business rules, calculations, and rollups in Dataverse Learning Path and in **Overview of building model-driven apps¹**.

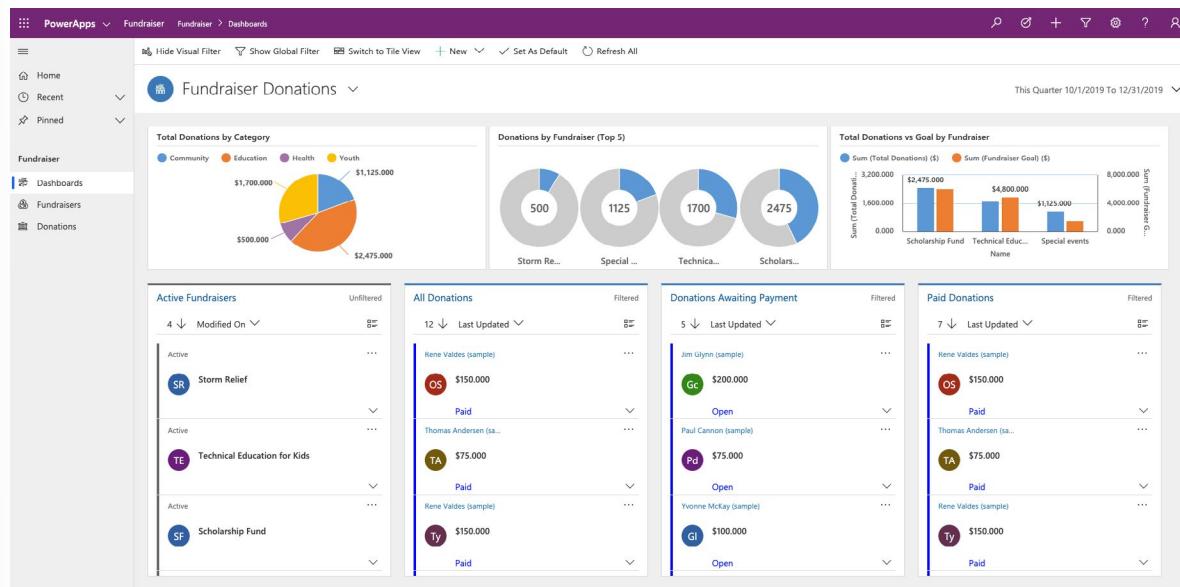
Exercise - Explore sample template apps

In **Power Apps²**, you can use a sample app to explore design possibilities. You'll also discover concepts that you can apply as you develop your own apps. Every sample app uses fictitious data to showcase a real-world scenario.

For more details, be sure to check out the documentation that's specific to each sample app.

¹ <https://docs.microsoft.com/powerapps/maker/model-driven-apps/model-driven-app-overview>

² <https://make.powerapps.com/?azure-portal=true>



Get sample apps

Before you can experiment with or edit the model-driven sample apps, you must set them up in a Dataverse database. First, create a trial environment and a database, and then select the **Include sample apps and data** check box.

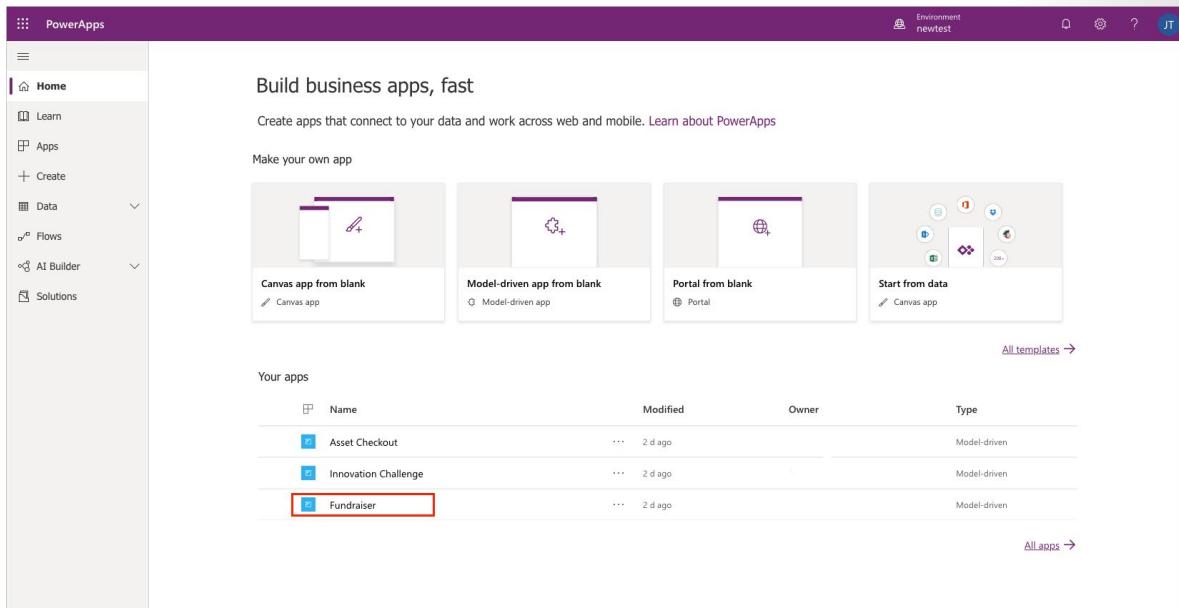
Important: By selecting the "Include sample apps and data" check box, you will install all available sample apps in your database. Sample apps are for educational and demonstration purposes. We don't recommend installing them in production databases.

Run a sample app

To run a sample app, follow these steps:

1. Sign in to **Power Apps**³ to see a list of available sample apps. For this example, select **Fundraiser**.

³ <https://make.powerapps.com/?azure-portal=true>



2. At the top of the page, select **Show Visual Filter** to show graphs and charts where you can see how donations to fundraisers are performing. In the next steps, you'll create a new fundraiser and submit a donation to that fundraiser.
3. On the left pane, under **Fundraiser**, select **Fundraisers**.
4. Select **+ New** to add a new fundraiser.
5. Under the **General** tab, enter the following information:
 - Name** - My Fundraiser
6. In the upper-right corner, select the drop-down arrow next to **Total Donations**.
7. From the drop-down menu, enter the following information:
 - Fundraiser Goal** - 500
8. On your keyboard, press **Enter**.
9. Select **Save & Close**.
10. On the left pane, under **Fundraiser**, select **Donations**.
11. Select the drop-down arrow to the right of **Other Activities**.
12. From the drop-down menu, select **Donation**.
13. Enter the following information:
 - **Subject** - My First Donation
 - **Donation Amount** - 100
 - **From** - Nancy Anderson (sample)
 - **Regarding** - My fundraiser
14. Select **Save & Close**.
15. On the left, select **Dashboards**.
16. Select **Show Visual Filter**.

17. Notice **Total Donations vs Goal by Fundraiser** shows your donation.

You have successfully run a sample model-driven app, added a new fundraiser, and added a donation to the fundraiser. While this is a quick look at an app and the related pieces, there is much more to learn about creating an app.

Summary

The goal of this module was to help you become familiar with the basics of Dataverse and creating model-driven apps.

Dataverse lets you store and manage data that is used by business applications. tables within Dataverse are used to store data.

To review, this module explained the following concepts:

- Dataverse data is stored in tables.
- You can use as many default tables as possible to quickly build apps.
- You can create new tables.
- Data is stored in a way that users can see it only if you grant them access.
- Sample apps and templates are available to help you learn and create your own apps.

Get started with Dataverse

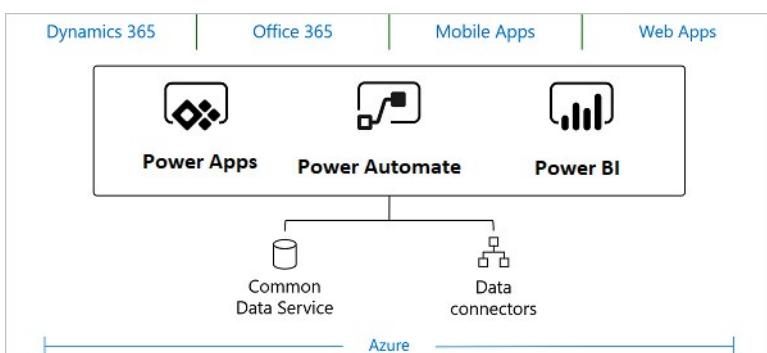
Introducing Dataverse

Dataverse lets you securely store and manage data used by business applications.

tables

Data within Dataverse is stored within a set of records called tables. A **table** is a set of records used to store data, similar to how a table stores data within a database.

Dataverse includes a base set of standard tables 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, from Excel, or from PowerQuery. App makers can then use Power Apps to build rich applications using this data.



Dynamics 365 applications, such as Dynamics 365 Sales, Service, and Marketing, use Dataverse to store and secure data used by the applications. This means you can build apps using Power Apps and Dataverse directly against your core business data already used within Dynamics 365 without the need for manual integration.

Note: Dynamics 365 Finance, Dynamic 365 Supply Chain Management, and Dynamics 365 Retail currently require the configuration of the Data Integrator to make your business data available within Dataverse.

For most organizations, it's a good idea to use the standard entities and attributes as they were intended. But to meet your business needs, you can extend the functionality of standard entities by creating one or more custom entities to store information that's unique to your organization.

Logic and validation

Entities within Dataverse can leverage rich server-side logic and validation to ensure data quality. You can also reduce repetitive code in each app that creates and uses data within a table.

- **Business rules:** Business rules validate data across multiple columns in a table, and provide warning and error messages, regardless of the app that's used to create the data.
- **Business process flows:** Business process flows guide users to ensure they enter data consistently and follow the same steps every time. Business process flows are currently supported only for model-driven apps.
- **Workflows:** Workflows automate business processes without requiring user interaction.

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

Security

Data in Dataverse is securely stored so that users can see it only if you grant them access. Role-based security, based on the Dynamics 365 system allows you to control access to entities for different users within your organization.

Learn about tables

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

Tables describe the kinds of data that is stored in the Dataverse database. Each table corresponds to a database table and each column (also known as an attribute) within a table represents a column in that table.

In Dataverse, metadata (data about data), is a collection of tables. Table metadata is what controls the kinds of records you can create and what kind of actions can be performed on them. When you use customization tools to create or edit tables, columns, and table relationships, you are editing this metadata.

The apps that your customers use to interact with the data in your environment depend on the table metadata, and they adapt as you customize the metadata.

When to use standard tables, and when to create new tables

Dataverse comes with a number of standard tables that support core business application capabilities. Each table also contains a number of metadata columns that represent common data that the system needs to store for that table. We recommend that you become very familiar with the catalog of standard tables, and use them where possible, because any applications written with standard tables will work as you expect in your environment without additional effort.

For minor changes, you might not have to create a custom table:

- To change the display name of a column, you can edit the table. You don't have to create a new table.
- You can't delete standard tables, but you can hide them. To hide a standard table, change the security role privileges for your organization to remove the Read privilege for that table. This will remove the table from most parts of the application.

If standard tables don't work for your business needs, and if they can't be edited to meet those needs, consider creating a new table, column, or table relationship. If a standard table almost meets your business needs, you can use it as the basis for a new table.

Table relationships

Table relationships define the different ways table records can be associated with records from other tables or the same table. Table relationships are metadata. They let queries retrieve related data very efficiently. Use table relationships to define the formal relationships that define the table or that most records can use.

When you look at the solution explorer you might think that there are three types of table relationships but actually there are only two, see below:

- **One-to-many relationships:** In a one-to-many (1:N) table relationship, many related table records are associated with a single primary table record in a parent/child relationship.
- **Many-to-many relationships:** In a many-to-many (N:N) table relationship, many table records are associated with many other table records. Records that are related through N:N table relationships are considered peers.

The N:1 (many-to-one) relationship type exists in the user interface because the designer shows you a view grouped by tables. 1:N relationships actually exist between tables and refer to each table as either a Primary/Current table or Related table. The related table, sometimes called the child table, has a lookup column that allows storing a reference to a record from the primary table, sometimes called the parent table. A N:1 relationship is just a 1:N relationship viewed from the related table.

Besides defining how records can be related to other records, 1:N table relationships also provide data to address the following questions:

- When I delete a record, should any records that are related to that record also be deleted?
- When I assign a record to a new owner, do I also have to assign all related records to the new owner?
- How can I streamline the data entry process when I create a new related record in the context of an existing record?
- How should people who view a record be able to view the related records?

table types

Before creating or editing tables in Dataverse, you should understand the different types of tables that you can create. After a custom table is created, the table type can't be changed.

Types of table owners

When you create a custom table, the options for ownership are *User or team owned*, or *Organization-owned*. After a table is created, you can't change the ownership.

- **User or team owned:** Actions that can be performed on these records can be controlled at the user level.
- **Organization-owned:** Access to the data is controlled at the organization level.

Activity tables

An *activity* is an action that a calendar entry can be made for. Activities have these characteristics:

- They have time dimensions (start time, stop time, due date, and duration) that help define when the action occurred or will occur.
- They have data (like a subject and description) that helps define the action that the activity represents.
- They can be opened, canceled, or completed. Several sub-status values will be associated with the *Completed* status of an activity to clarify how the activity was completed.

Activity tables can be owned only by a user or team. They can't be owned by an organization.

The following default activity tables are available:

- **Appointment:** A commitment representing a time interval that has start/end times and duration.

- **Email:** An activity that's delivered by using email protocols.
- **Fax:** An activity that tracks the call outcome and number of pages for a fax. The activity can optionally store an electronic copy of the document.
- **Letter:** An activity that tracks the delivery of a letter. The activity can store an electronic copy of the letter.
- **Phone Call:** An activity that tracks a telephone call.
- **Recurring Appointment:** The master appointment of a recurring appointment series.
- **Task:** A generic activity representing work that must be done.

Custom activity tables

You can create new custom activity tables. The metadata values of activity tables differ from the metadata values of other tables. For example, the **Primary** column is set to **Subject**.

Business Rules

Business rules provide a simple interface to implement and maintain fast-changing and commonly used rules. The business rules defined for a table apply to both canvas apps and model-driven apps if the table is used in the app.

By combining conditions and actions, you can do any of the following with business rules:

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

Differences between canvas and model-driven apps

Model-driven apps can use all actions available on business rules, however not all business rule actions are available for canvas apps at this time. The following actions are not available on Canvas apps:

- Show or hide columns
- Enable or disable columns
- Create business recommendations based on business intelligence.

A great reference for a deep dive into this lesson is in this **learning path**⁴

⁴ <https://docs.microsoft.com/en-us/learn/patterns/create-relationships-common-data-service/>

Exercise - Create a Dataverse table

In this unit, you'll create a table and then customize key components, like columns, relationships, views, and forms. You'll learn how to:

- Create a custom table.
- Add custom columns to your table.
- Add a table relationship.
- Customize a view.
- Customize a form.

The tutorial follows the Contoso company, which is a pet grooming business that grooms dogs and cats. Contoso needs an app for client and pet tracking that can be used by employees on a variety of devices.

Create a custom table

Sign in to **Power Apps**⁵ and follow these steps to create a new custom table.

1. In the left navigation pane, expand **Data**, select **tables**, and then select **+ New table**.
2. Under New table, enter the following:
 - **Display name:** Pet
3. In the Primary column section, enter the following:
 - **Display name:** Pet Name
4. At the bottom, click **Create**.

You will notice in our example, the new table and primary column begins with **cree0_**. Additional columns created for this table will also begin with **cree0**, this is specific to our demo environment. When testing in your own environment this may look different.

Add and customize columns

1. In the list of tables, select the **Pet** table that you created in the previous section.
2. On the **columns** tab, on the table designer toolbar, select **Add column**.
3. In the **column properties** pane, enter the following values:
 - **Display name:** Species
 - **Data type:** Choice
 - **Choice set:** + New choice
 - **Searchable:** Yes
4. Create the choice set:
 1. Replace *New choice* with *Dog*.
 2. Select **Add new item**.
 3. Replace *New choice* with *Cat*.
 4. Select **Save**.

⁵ <https://make.powerapps.com>

Species (1)

Display name *

Name *

[View more](#)

Items (2)

Dog	...
Cat	...

[Add new item](#)

Save Cancel

5. Make sure **Searchable** is selected, and then select **Done**.
6. On the table designer toolbar, select **Add column**.
7. In the **column properties** pane, enter the following values, and then select **Done**:
 - **Display name:** Breed
 - **Data type:** Text
 - **Searchable:** Yes
8. On the table designer toolbar, select **Add column**.
9. In the **column properties** pane, enter the following values, and then select **Done**:
 - **Display name:** Appointment date
 - **Data type:** Date and Time
 - **Searchable:** Yes
10. Select **Save table**.

Add a relationship

1. On the **Relationships** tab, on the table designer toolbar, select **Add relationship**, and then select **Many-to-one**.
2. In the right pane, in the **Related** list, select **Account**.
3. Select **Done**.
4. Select **Save table**.

Notice that when you add a many-to-one relationship, an **Account** column of the **Lookup** data type is automatically added to your list of columns on the **columns** tab.

Display name ↑ ↴	Name ↴	Data type ↴	Type ↴	Custom... ↴	Required ↴	Search... ↴
Account	... cree0_account	Lookup	Custom	✓	✓	
Appointment date	... cree0_appoint...	Date an...	Custom	✓	✓	
Breed	... cree0_breed	Text	Custom	✓	✓	
Pet Name Primary Field	... cree0_pet	Text	Custom	✓	✓	✓
Species	... cree0_species	Option ...	Custom	✓	✓	

Customize a view

- On the **Views** tab, right-click **Active Pets** view and select **Open Link in New Tab**. If you don't see the **Active Pets** view, select **Remove filter**.
- In the view designer, select **+ View column**, select the following columns, and then select **OK**:
 - Account
 - Appointment date
 - Breed
 - Species
- Select the **Created On** column, select **Remove**.
- To arrange the columns, select the column to move, and then select **Move Left** or **Move Right** until your view looks like this. You could also simply drag and drop the columns to arrange the order as well.

- On the view designer toolbar, select **Save**.
- Select **Publish**.

Customize the main form

2. In the left navigation pane, expand **Data**, select **tables**, and then select **Pet**.
3. On the **Forms** tab, select **Information** next to the **Main** form type to open the form editor.

The screenshot shows the PowerApps Entities screen for the Pet table. The left sidebar has sections for Home, Learn, Apps, Create, Data, Entities, Option Sets, Dataflows, and Export to data lake (preview). The main area shows Fields, Relationships, Business rules, Views, Forms, Dashboards, Charts, Keys, and Data. Under Forms, there are three entries: 'Information' (Main, Custom), 'Information' (Card, Custom), and 'Information' (QuickViewForm, Custom). The 'Information' entry under 'Main' is highlighted with a red box.

4. In the form editor, drag the **Species**, **Breed**, **Appointment date**, and **Account** columns from the **column Explorer** pane to the **General** section of the form canvas, so that the form looks like this.

The screenshot shows the PowerApps Form designer for a 'New Pet' form. The left sidebar shows the 'Fields' pane for the Pet entity, with various fields listed and a checkbox for 'Show only unused fields'. The main canvas displays the 'General' section of the form with fields for Pet Name, Owner (MOD Administrator), Species, Breed, Appointment date, and Account. The right sidebar contains settings for the form, including Entity (Pet), Display Name (Information), Description (A form for this entity), Max Width (pixels) (1,920), and a checked 'Show image' option.

5. Select **Save**.
6. Select **Publish**.
7. Click the back arrow in your browser to close the form designer.

Exercise - Create a table and import data into your Dataverse

You can import data into your Dataverse database in bulk from Microsoft Excel or CSV files.

Every table has required columns that must exist in your input file. We recommend that you create a template. A template will save you time and effort. First, export data from the table. You'll use the same file (updated with your data) to import data into the table.

Create a file template

You can do a one-time data export from a standard table or a custom table, and you can export data from more than one table at a time. If you export data from more than one table, each table is exported into its own Microsoft CSV file. In this example, you'll see how to export the **Pet** table but remember you could select several tables to export if you would like.

1. On [powerapps.com](https://make.powerapps.com)⁶, in the left navigation pane, expand **Data**, select **tables**, and then select **Export data**.
2. Select the **Pet** table, and then select **Export data**.
3. After the export is finished, select **Download exported data**, and save the file.

Copy data into your template

When you add data to a template file, you must make sure the data is unique. You can use either *primary keys* or *alternate keys*.

1. Open the CSV file that you created in the previous section.
2. Add at least one new row of data but you only need to add information to the following columns below.
 - **Appointment date**
 - **Breed**
 - **Pet Name**
 - **Species**

The screenshot shows a Microsoft Excel spreadsheet titled "cree0_pets". The data is organized into columns with the following headers:

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	cree0_appointmentdate	cree0_breed	importsequencenumber	cree0_petid	cree0_petname	cree0_species	statuscode	statecode	timezoneruleversionnumber	utcconversiontimezonecode	versionnumber	cree0_account	owningbusinessunit
2	11/14/19	Goldendoodle			Chewy	Dog							
3													
4													
5													
6													
7													
8													
9													
10													
11													

3. Save the file.

Import the file

1. In the left navigation pane, expand **Data**, and then select **tables**.
2. Select the **Pet** table, select the drop down arrow next to Get Data, and then select **Get data from Excel**.
3. Click **Upload** and select the file that you just updated and saved from the previous section.
4. After the file is uploaded, click **Map columns**.

⁶ <https://make.powerapps.com/>

5. Set the following Pet columns Source values to **None**.

- Import Sequence Number
- Owning Business Unit
- Pet
- Status Reason Value
- Status Value
- Time Zone Rule Version Number
- UTC Conversion Time Zone Code
- Version Number

Pet fields	Source values
Account	cree0_account
Appointment date	cree0_appointmentdate
Breed	cree0_breed
Import Sequence Number	Not set
Owning Business Unit	Not set
Pet	Not set
Pet Name *	cree0_petname
Species Value	cree0_species
Status Reason Value	Not set
Status Value	Not set
Time Zone Rule Version Number	Not set
UTC Conversion Time Zone Code	Not set
Version Number	Not set

6. In the upper right, click **Save changes**.

Name	File	Mapping status
Pet	cree0_pets.csv	Upload ⚠️ Mapping warnings exist

You will notice under Mapping status, it states "Mapping warnings exist". The reason for this mapping status is because we set some of the columns to None or Not. This is fine because we didn't want to include those columns so this warning can be ignored.

-
7. In the upper right, click **Import**.

After the data has been successfully imported, you'll see the total number of inserts and updates. Now let's go take a quick look at the imported data

8. On the left, select **tables**.
9. Select the **Pet** table.

10. On the table designer toolbar, select **Data**.

Notice that the Account column is empty, this is because when you updated the Excel export file and update columns you were not instructed to update this column. The reason you were not instructed to update the Account column is because you cannot set a lookup value when importing data from Excel, this must be done from Power Apps.

Scenario

The current sales process for your company is manual and updates are only provided each Friday. To simplify this process, minimize the opportunity for mistakes, and improve visibility, you have decided you want to create a new app to track sales leads and automatically calculate the forecasted revenue. You want to use Common Data Service to store the list of potential customers.

Use Dataverse to store data

In this exercise, you will use Dataverse to store the list of potential customers for your app.

Creating a Custom table

1. Go to the **Power Apps home page**⁷ and sign in to Power Apps.
2. On the menu, expand **Data** and Select **tables**.
3. Select **New table**.
4. Enter the following information:
 - **Display name:** Prospects
5. In the Primary Name column section, enter the following information:
 - **Display name:** Prospect Name
6. Select **Create**.
7. Select **Add column**.
8. Enter the following information:
 - **Display name:** Stage
 - **Data Type:** Choice
9. For **choice**, Select the dropdown and select **New choice**. Enter the following information and Select **Save**.
 - **Display name:** Prospect Stage

⁷ <https://powerapps.microsoft.com/?azure-portal=true>

- **Name:** prospectstage
 - Add the following items/options: **Lead, Opportunity, Won, Lost**
10. Once the choice has saved, continue entering the following information for the new column:
- **Default value:** Lead
 - Change **Required** dropdown to **Required**
11. Select **Done**.
12. Select **Add column**.
13. Enter the following information and then Select **Done**.
- **Display name:** Contract Amount
 - **Data Type:** Currency
14. Select **Add column**.
15. Enter the following information and then Select **Done**.
- **Display name:** Probability
 - **Data Type:** Whole Number
16. Select **Save table**, in the bottom-right corner.
17. Select **Add column**.
18. Enter the following information and then Select **Done**.
- **Display name:** Forecasted Revenue
 - **Data Type:** Currency
 - Select **+Add** for Calculated or Rollup
 - Select **+Calculation**
19. On the popup, Select **Save**.
20. A new browser window will open. Select **Add action** near the bottom.
21. Enter the following formula, but do not copy and paste, type it in as your column names will not be exactly the same as the example below since the **crXXX_** will be specific numbers and letters that define your environment. The formula entry will auto-suggest you options as you start to type the column names.
- ```
crXXX_contractamount * (crXXX_probability / 100)
```
22. Select the checkmark to save your changes. You may need to scroll right to see it.
23. Select **SAVE AND CLOSE**.
24. Select **Done**.

## Add a Business Rule

1. On the table designer toolbar, Select **Business rules**.
2. Select **Add business rule**, a new browser tab will open.
3. Select on the **Condition New Condition** from the design pane.
4. In the right-hand pane, for **field** choose **Contract Amount**.

5. For the Operator, choose **contains data**.
6. Select **Apply**.
7. In the right-hand pane, Select **Components**.
8. Select and hold **Set Business Required** and drag to the plus symbol to the right of the purple checkbox in the design pane.
9. In the right-hand pane, for **field** choose **Probability**.
10. For the Status, choose **Business Required**.
11. Select **Apply**.
12. In the top left of the screen, Select the dropdown next to **Prospects New business rule** and set the **Business rule name** to **Make Probability Required**.
13. Select **Save** in top-right corner of screen.
14. Select **Activate** to activate the rule.
15. Select **Activate** to confirm activation.
16. Close the browser tab.
17. Now back on the table management screen, Select **Done**.

## Importing Data from an Excel file

You will use the Excel spreadsheet named **Prospects**<sup>8</sup> for this exercise. Open the link, and select the Download button and save it locally.

1. Open the file. Notice the "Stage" column is empty, you will need to look these up and enter them manually.
2. Go to the **Power Apps home page**<sup>9</sup> and sign in to Power Apps.
3. Go back to your Excel file and in the Stage column enter the values as below:
  - **Contoso Flooring**: Won.
  - **Fabrikam Inc**: Won.
  - **Adventure Works**: Lead.
  - **Adatum**: Lead.
  - **VanArsdel**: Lost.
  - **Relecloud**: Opportunity.
4. Save and close the Excel file.
5. Continue with the **Prospects** table.
6. Select the drop-down arrow to the right of **Get data** and select **Get data from Excel**.

[!NOTE]  
If you receive an error when selecting the **Get data from Excel** option, then your Power Apps license does not allow for importing Excel data.
7. Select **Upload**, locate the Prospects Excel file, and Select **Open**.

<sup>8</sup> <https://github.com/MicrosoftDocs/mslearn-developer-tools-power-platform/blob/master/power-apps/Prospects.zip?azure-portal=true>

<sup>9</sup> <https://powerapps.microsoft.com/?azure-portal=true>

8. Select **Map columns**. Map the following Prospect columns to the associated Source values:
  - **Contract Amount**: *ContractAmount*
  - **Prospect Name**: *Name*
  - **Stage Value**: *Stage*
  - **Probability**: *Probability*
9. Select **Save Changes** at the top.
10. Select **Import**.
11. Select **tables**.
12. Select the **Prospects** table and Select **Data**.
13. Ensure that the data has successfully imported.

## Summary

Congratulations on using Dataverse, and creating your first table!

Dataverse lets you securely store and manage data that's used by business applications. Standard and custom tables within Dataverse provide a secure and cloud-based storage option for your data.

Let's review what you've learned:

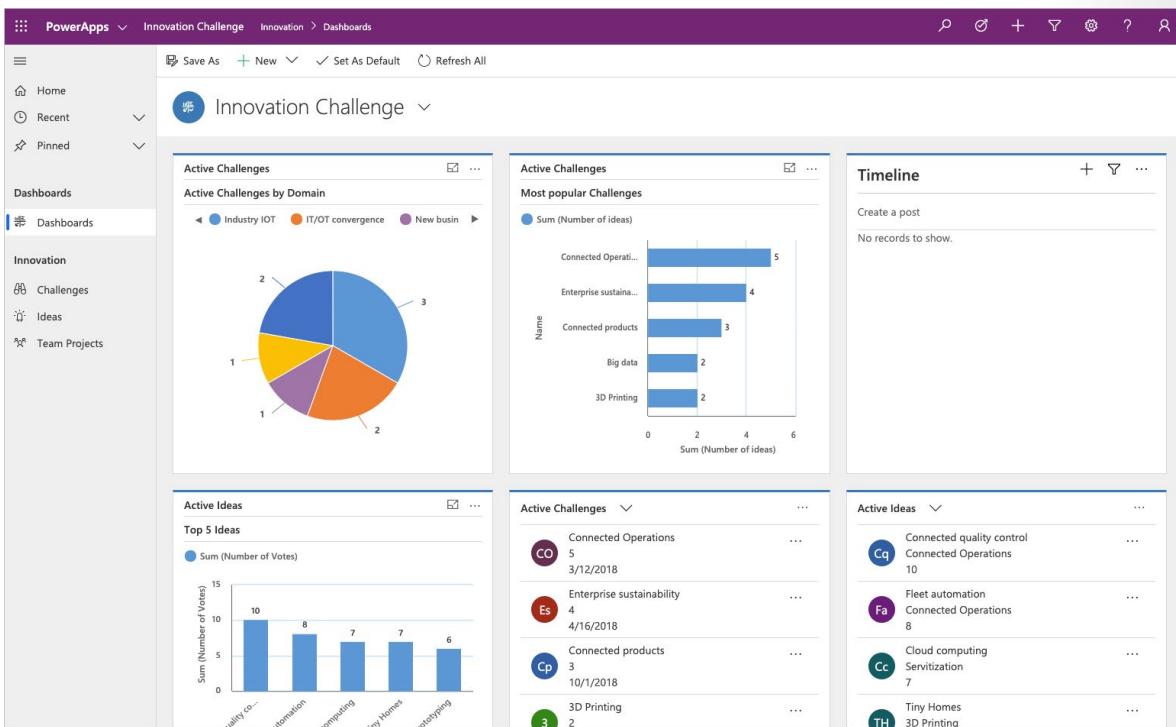
- Dataverse stores data in tables.
- If you can, it's preferable to use as many default tables as possible, so that your system is easy to maintain.
- Creating new tables is easy.
- The best way to get data into a table is to export a template, add data, and upload it.

# Get started with model-driven apps in Power Apps

## Introducing model-driven apps

Model-driven app design is an approach that focuses on adding dashboards, forms, views, and charts to your apps. With little or no code, you can build apps that are simple or very complex.

In canvas apps, the app maker has total control over the app layout. In model-driven apps, on the other hand, much of the layout is determined by the components you add. The emphasis is more on quickly viewing your business data and making decisions instead of on intricate app design.



## The approach to making model-driven apps

Model-driven apps have three design phases:

1. Model your business data
2. Define your business processes
3. Build the app

## Model your business data

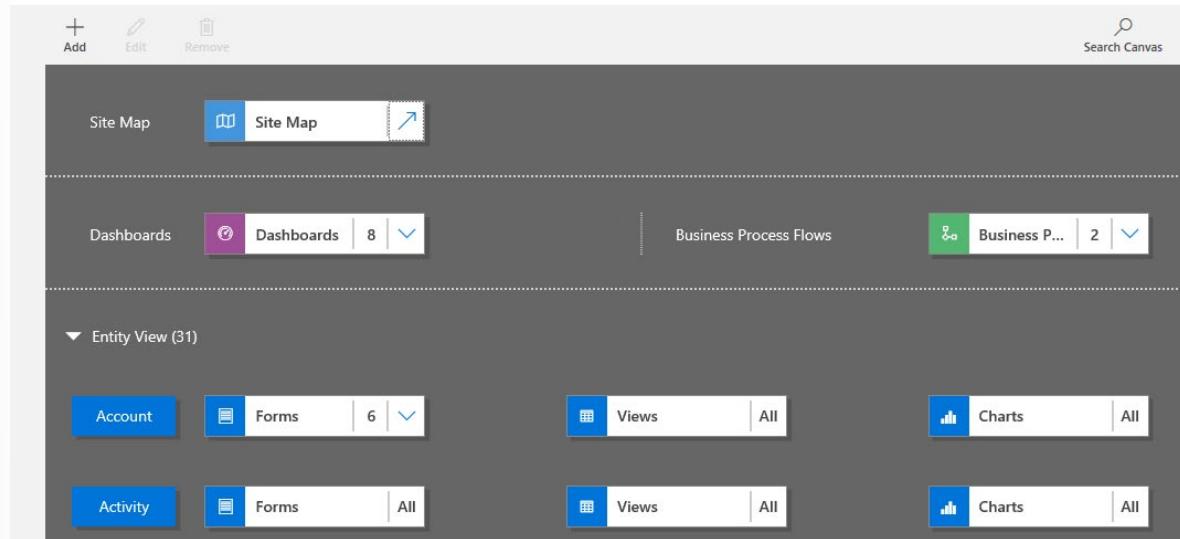
Model-driven design uses metadata-driven architecture so that designers can customize apps without writing code. To model business data, you determine what data the app will need and how that data will relate to other data. Metadata means *data about data* and it defines the structure of the data stored in Dataverse.

## Define your business processes

Defining and enforcing consistent business processes is a key aspect of model-driven app design. Consistent processes help ensure that your app users can focus on their work and not worry about having to remember to perform a set of manual steps. Processes can be simple or complex, and they often change over time.

## Build the app

After modeling data and defining processes, you build your app by selecting and setting up the components you need in the App Designer.



## Building blocks of model-driven app

A model-driven app consists of several components that you select by using the App Designer. The components and component properties become the metadata. Let's look more closely at these components.

### Data

The data components determine what data the app will be based upon.

| Component | Description                                                                                                                                                                                                                                        | Designer       |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| table     | tables are items with properties that you track. Examples include contacts and accounts. Many standard tables are available. You can customize a non-system standard table (or production table). You can also create a custom table from scratch. | table designer |

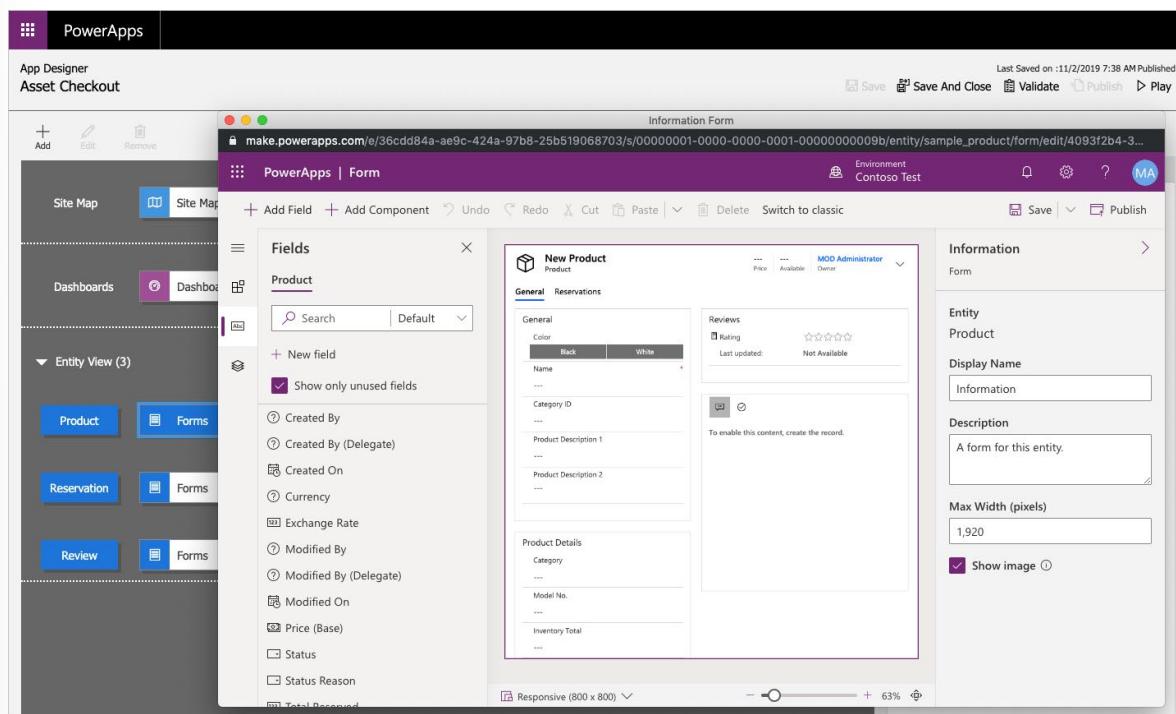
| Component     | Description                                                                                                                                                                                                                                                                                                                                                                                | Designer       |
|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------|
| column        | columns are properties that are associated with a table and help define that table. A column is defined by a data type, which determines the type of data that can be entered or selected. Examples of data types include text, number, date and time, currency, and lookup (which creates a relationship with another table). columns are typically used with forms, views, and searches. | table designer |
| Relationship  | Relationships define how tables can be related to each other. There are 1:N (one-to-many), N:1 (many-to-one), and N:N (many-to-many) relationships. For example, adding a lookup column to a table creates a new 1:N relationship between the two tables and lets you add that lookup column to a form.                                                                                    | table designer |
| Choice column | This type of column shows a control that lets the user select among predefined options. Each option has a number value and a label. Choice columns can require either a single value or multiple values.                                                                                                                                                                                   | table designer |

## User interface

The user interface components determine how users will interact with the app.

| Component | Description                                                                                     | Designer          |
|-----------|-------------------------------------------------------------------------------------------------|-------------------|
| App       | Apps determine the app fundamentals, like components, properties, the client type, and the URL. | App designer      |
| Site map  | A site map specifies the navigation for your app.                                               | Site map designer |

| Component | Description                                                                                                                                                                                                                                                                                                                        | Designer      |
|-----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|
| Form      | Forms include a set of data entry columns for a given table. This set of data entry columns matches the items that your organization tracks for the table. One example is a set of data entry columns where users enter relevant information to track a customer's previous orders together with specific requested reorder dates. | Form designer |
| View      | Views define how a list of records for a specific table appears in your app. A view defines the columns shown, the width of each column, the sort behavior, and the default filters.                                                                                                                                               | View designer |



## Logic

The logic components determine what business processes, rules, and automation the app will have. Microsoft Power Apps makers use a designer that's specific to the type of process or rule.

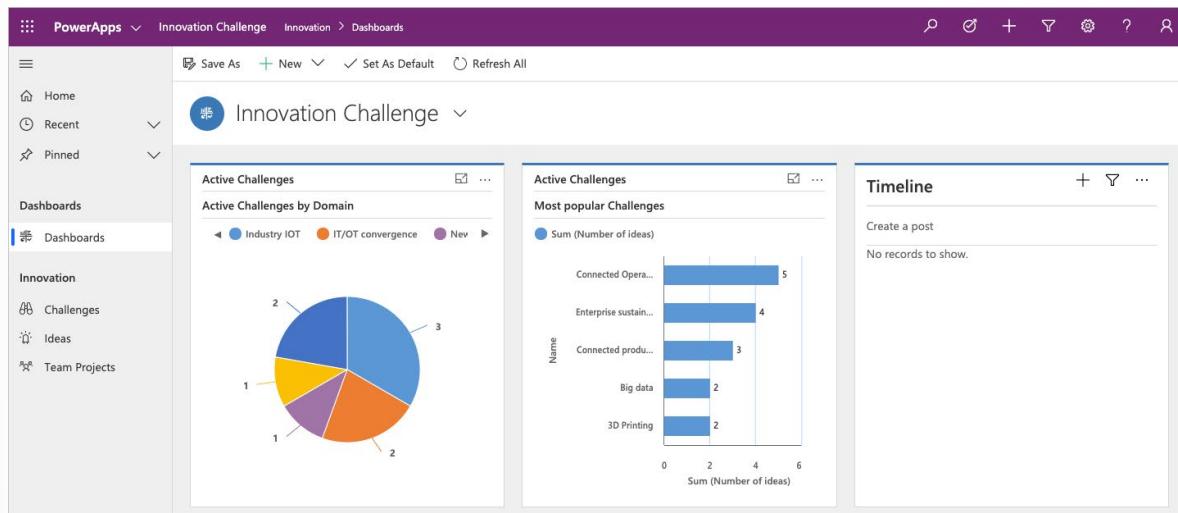
| Type of logic         | Description                                                                                                                                                                                                                                                                                   | Designer                       |
|-----------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------|
| Business process flow | Business process flows walk users through a standard business process. Use a business process flow if you want everyone to handle customer service requests the same way. Or you can use a business process flow to require staff to gain approval for an invoice before submitting an order. | Business process flow designer |
| Workflow              | Workflows automate business processes without a user interface. Designers use workflows to initiate automation that doesn't require any user interaction.                                                                                                                                     | Workflow designer              |
| Actions               | Actions are a type of process that lets you manually invoke actions, including custom actions, directly from a workflow.                                                                                                                                                                      | Process designer               |
| Business rule         | Business rules apply rules or recommendation logic to a form to set column requirements, hide columns, validate data, and more. App designers use a simple interface to implement and maintain fast-changing and commonly used rules.                                                         | Business rule designer         |
| Flows                 | Power Automate is a cloud-based service that lets you create automated workflows between apps and services to get notifications, sync files, collect data, and more.                                                                                                                          | Power Automate                 |

## Visualization

The visualization components determine what type of data and reporting the app will show.

| Component | Description                                                                                                             | Designer           |
|-----------|-------------------------------------------------------------------------------------------------------------------------|--------------------|
| Chart     | Charts are individual graphical visualizations that can appear in a view or a form or that can be added to a dashboard. | Chart designer     |
| Dashboard | Dashboards show one or more graphical visualizations that provide an overview of actionable business data.              | Dashboard designer |

| Component                   | Description                                                                                                                                           | Designer                                                          |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| Embedded Microsoft Power BI | Power BI adds embedded Power BI tiles and dashboards to your app. Power BI is a cloud-based service that provides business intelligence (BI) insight. | A combination of chart designer, dashboard designer, and Power BI |



## Advanced model-driven apps

Solution Explorer is used to make advanced model-driven apps. By using the navigation pane on the left side of the tool, you can navigate a hierarchy that consists of all app components.

To access the classic Solution Explorer, you must first select a Solution then select **Switch to classic**.

The screenshot shows the PowerApps interface with the title "Solutions > Default Solution". On the left, there's a navigation pane with "Solutions" selected. The main area displays a table of app components.

| Display name        | Name                  | Type             | Managed... | Modified | Owner | Status |
|---------------------|-----------------------|------------------|------------|----------|-------|--------|
| Application Ribbons | ...                   | Client Extension | ⋮          | -        | -     | -      |
| Site Map            | ...                   | Client Extension | ⋮          | 1 wk ago | -     | -      |
| Account             | ... account           | Entity           | ⋮          | -        | -     | -      |
| Account Manager     | ... Account Manager   | Connection Role  | ⋮          | 1 wk ago | -     | -      |
| Account Reconnect   | ... Account Reconnect | Email Template   | ⋮          | 1 wk ago | -     | -      |
| Account Summary     | ... Account Summary   | Report           | ⋮          | 4 d ago  | -     | -      |

The screenshot shows the PowerApps Entity list screen. On the left, there's a navigation pane titled "Solution: Default Solution" with categories like Information, Components, Entities, Option Sets, Client Extensions, Web Resources, Processes, and more. The "Entities" category is selected. The main area displays a table of entities:

| Display Name                     | Name                     | Schema Name              | State   | Customizable | Audit Status   | Description                                |
|----------------------------------|--------------------------|--------------------------|---------|--------------|----------------|--------------------------------------------|
| Account                          | account                  | Account                  | Managed | True         | Disabled       | Business that represents a customer or     |
| Action Card                      | actioncard               | ActionCard               | Managed | True         | Disabled       | Action card entity to show action cards.   |
| Activity                         | activitypointer          | ActivityPointer          | Managed | True         | Non Applicable | Task performed, or to be performed, by     |
| Address                          | customeraddress          | CustomerAddress          | Managed | True         | Disabled       | Address and shipping information. Use      |
| Appointment                      | appointment              | Appointment              | Managed | True         | Disabled       | Commitment representing a time interv      |
| Attachment                       | activitymimeattachment   | ActivityAttachment       | Managed | True         | Disabled       | MIME attachment for an activity.           |
| Attribute                        | attribute                | Attribute                | Managed | True         | Disabled       |                                            |
| Business Unit                    | businessunit             | BusinessUnit             | Managed | True         | Disabled       | Business, division, or department in the   |
| Challenge                        | sample_challenge         | sample_challenge         | Managed | True         | Disabled       |                                            |
| Challenge Management Process     | sample_challengeman...   | sample_challengeman...   | Managed | True         | Disabled       | Base entity for process Challenge Mana     |
| Channel Access Profile           | channelaccessprofile     | ChannelAccessProfile     | Managed | True         | Enabled        | Information about permissions needed       |
| Channel Access Profile Rule      | channelaccessprofiler... | ChannelAccessProfileR... | Managed | True         | Disabled       | Defines the rules for automatically asso   |
| Channel Access Profile Rule Item | channelaccessprofiler... | ChannelAccessProfileR... | Managed | True         | Disabled       | Defines the rule items of a profile rule s |

## Design model-driven apps

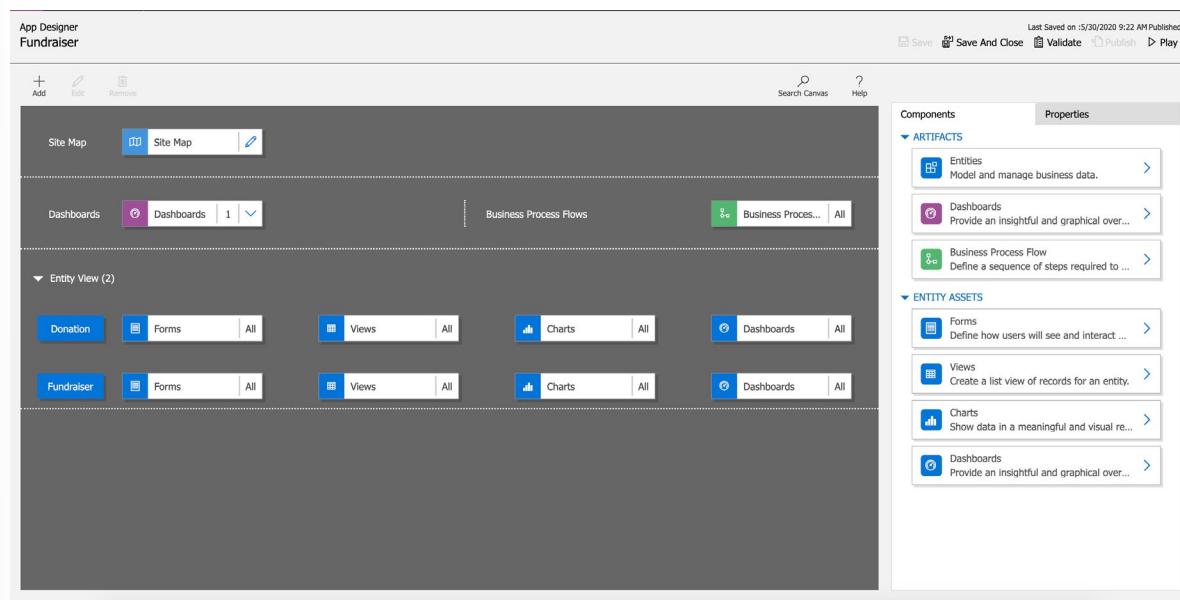
As an App Maker, before you begin building your Power Apps solution, it's recommended to go through a design process.

When designing your Power Apps solution, there are several different factors to consider:

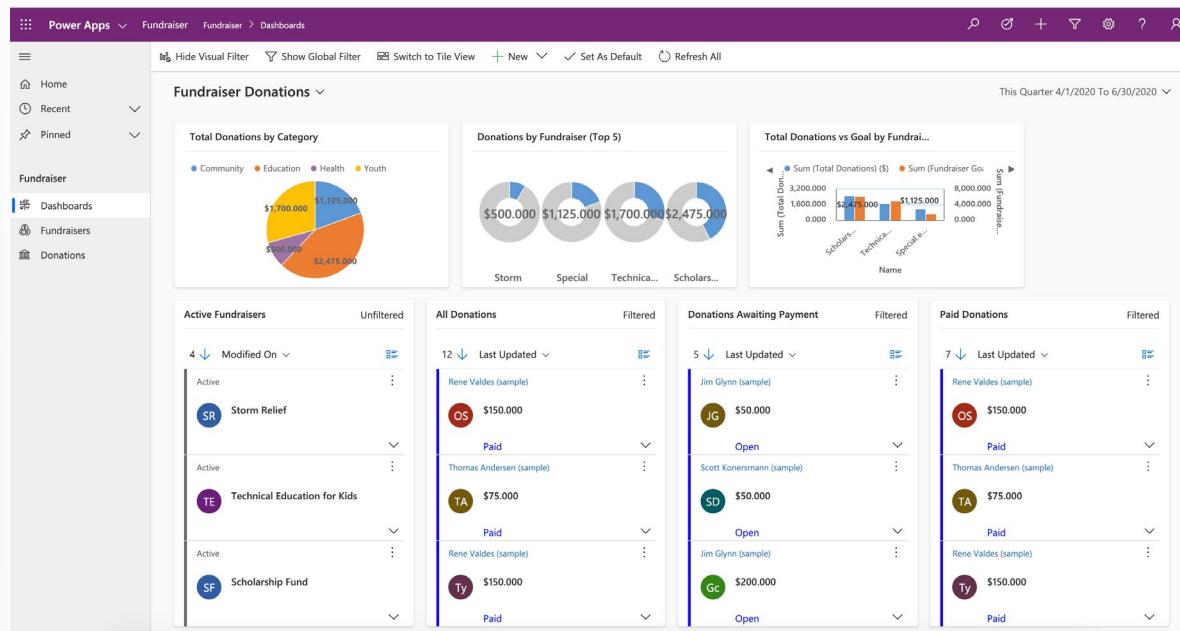
- Business requirements
- Data Model
- Business Logic
- Output

By going through a simple design process, you can flush out any minor issues before they become a larger problem once the app is in production.

Here is a quick look at the App Designer for an example Model-driven app called "Fundraiser."



When the app is put into Play Mode, it looks completely different.



## Understand the needs of the user

With Model-driven apps, the name says it all. Your primary design goal is to get your Dataverse data model in order. With that in place, you can connect Power Apps, and a Model-driven app will be created for you from that model.

Model-driven apps are created using the App Designer. You will choose the tables, dashboards, Business Process flows, forms, and other components that you want to make available in your app, and then the app will be created for you. This means you will need to spend more time understanding what your user needs than how it is going to look.

## Business Requirements

The first step in the process is to understand your business requirements. Work with the app stakeholders to consider your security, accessibility, data, and design needs.

For security, the Dataverse has a robust security model. You will want to consider how securing your app's data affects your app and what security model best supports your business needs. There are lots of options available, including hierarchy security, row-level security, to name a few. You will need to confirm your data is secured to meet your needs, and then your app will honor that security. For more information, see **Security in Dataverse**<sup>10</sup>.

During this process, you will also want to identify any government regulations or authentication/authorization requirements (if applicable). You may want to implement multi-factor authentication but will need to think about how this will affect users connecting to your application. You don't necessarily have to have all the answers to your questions here; you just want to flush out all of the requirements.

Finally, does your app need to be available when the user is disconnected from the internet? This is called Offline Mode and is supported by the Dataverse and Model-driven apps when using iOS or Android clients. It does require additional design considerations. For more information, see **Set up mobile offline synchronization**<sup>11</sup>.

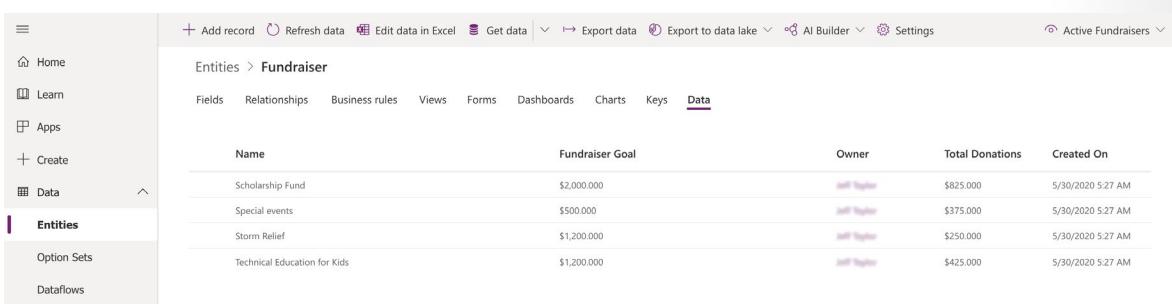
## Data Model

As you begin the data modeling process, there are a couple of important questions to ask yourself:

- What type of data will your solution be storing and or collecting?
- How will this data relate or coincide with the other data you are working with?

These questions are important when designing a model-driven application because of how model-driven applications function. Remember, model-driven applications use a metadata-driven architecture. This means a large portion of the model-driven app is based on how your data is modeled, and there is no need to write custom code to alter the app design. To expand on this a little further, when thinking about Metadata this simply means "data about data" and this data defines the structure stored in the system.

You can view the app metadata by reviewing the table in the Dataverse.



The screenshot shows the Microsoft Power Platform Data view interface. On the left, there's a navigation sidebar with Home, Learn, Apps, Create, Data (selected), Entities (selected), Option Sets, and Dataflows. The main area shows the 'Fundraiser' entity details under 'Entities > Fundraiser'. At the top right are various actions: Add record, Refresh data, Edit data in Excel, Get data, Export data, Export to data lake, AI Builder, and Settings. Below that is a dropdown for 'Active Fundraisers'. The central part of the screen displays a table with columns: Name, Fundraiser Goal, Owner, Total Donations, and Created On. The data in the table is as follows:

| Name                         | Fundraiser Goal | Owner       | Total Donations | Created On        |
|------------------------------|-----------------|-------------|-----------------|-------------------|
| Scholarship Fund             | \$2,000,000     | Jill Taylor | \$825,000       | 5/30/2020 5:27 AM |
| Special events               | \$500,000       | Jill Taylor | \$375,000       | 5/30/2020 5:27 AM |
| Storm Relief                 | \$1,200,000     | Jill Taylor | \$250,000       | 5/30/2020 5:27 AM |
| Technical Education for Kids | \$1,200,000     | Jill Taylor | \$425,000       | 5/30/2020 5:27 AM |

You can also view the app metadata by putting the app in Play mode.

<sup>10</sup> <https://docs.microsoft.com/power-platform/admin/wp-security?azureportal=true>

<sup>11</sup> <https://docs.microsoft.com/dynamics365/mobile-app/setup-mobile-offline-for-admin?azureportal=true>

The screenshot shows a Power Apps interface for managing fundraisers. The main view displays a table titled 'Active Fundraisers' with the following data:

| Name                         | Fundraiser Goal | Owner       | Total Donations | Created On        |
|------------------------------|-----------------|-------------|-----------------|-------------------|
| Scholarship Fund             | \$2,000.00      | Jeff Taylor | \$825.000       | 5/30/2020 5:27 AM |
| Special events               | \$500.000       | Jeff Taylor | \$375.000       | 5/30/2020 5:27 AM |
| Storm Relief                 | \$1,200.000     | Jeff Taylor | \$250.000       | 5/30/2020 5:27 AM |
| Technical Education for Kids | \$1,200.000     | Jeff Taylor | \$425.000       | 5/30/2020 5:27 AM |

In the example above, for the Fundraiser table, there are several pieces of metadata being collected, such as:

- Name
- Fundraiser Goal
- Owner
- Total Donations
- Created On

Each solution you develop and deploy will have its own set of metadata to collect. This basic understanding of metadata is important as you continue the design process and modeling your app data.

As you think about your data model also think about column types. When adding columns to your table in the Dataverse, the column type you choose will determine how users enter and view that in your Model-driven app. Option sets show as dropdowns, currency shows with currency symbols, while decimal numbers don't. These little changes in the table can have a profound effect on how your user experiences your app.

**Note:** If a column type needs to be changed to a different column type, (i.e. text column to an choice), then you will need to delete that column and re-create with the correct column type. This will cause you to lose any data associated with that columnn.

The screenshot shows the 'Fields' tab for the 'Fundraiser' entity in the Microsoft Dataverse. The table displays the following fields:

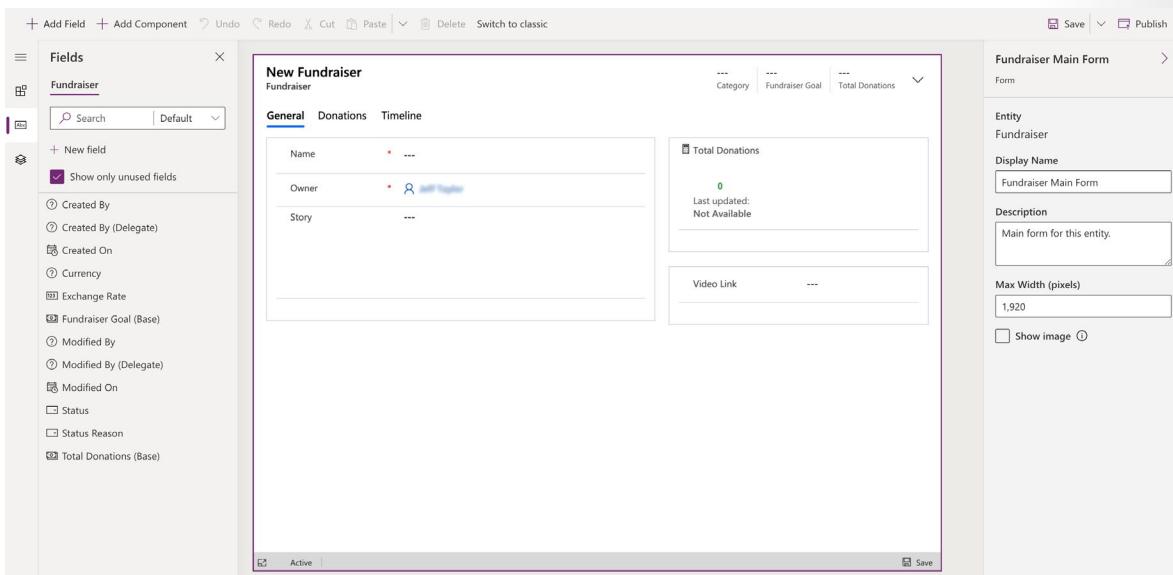
| Display name           | Name                       | Data type         | Type     | Customizable | Required | Searchable |
|------------------------|----------------------------|-------------------|----------|--------------|----------|------------|
| Category               | sample_category            | Option Set        | Managed  | ✓            | Optional | ✓          |
| Created By             | createdby                  | Lookup            | Standard | ✓            | Optional | ✓          |
| Created By (Delegate)  | createdonbehalfby          | Lookup            | Standard | ✓            | Optional | ✓          |
| Created On             | createdon                  | Date and Time     | Standard | ✓            | Optional | ✓          |
| Currency               | transactioncurrencyid      | Lookup            | Custom   | ✓            | Optional | ✓          |
| Exchange Rate          | exchangerate               | Decimal Number    | Custom   | ✓            | Optional | ✓          |
| Fundraiser             | sample_fundraiserid        | Unique Identifier | Standard | ✓            | Required | ✓          |
| Fundraiser Goal        | sample_fundraisergoal      | Currency          | Managed  | ✓            | Optional | ✓          |
| Fundraiser Goal (Base) | sample_fundraisergoal_base | Currency          | Managed  | ✓            | Optional | ✓          |
| Import Sequence Number | importsequencenumber       | Whole Number      | Standard | ✓            | Optional | ✓          |
| Modified By            | modifiedby                 | Lookup            | Standard | ✓            | Optional | ✓          |
| Modified By (Delegate) | modifiedonbehalfby         | Lookup            | Standard | ✓            | Optional | ✓          |
| Modified On            | modifiedon                 | Date and Time     | Standard | ✓            | Optional | ✓          |
| Name                   | sample_name                | Text              | Managed  | ✓            | Required | ✓          |
| Owner                  | ownerid                    | Owner             | Standard | ✓            | Required | ✓          |
| Owning Business Unit   | owningbusinessunit         | Lookup            | Standard | ✓            | Optional |            |
| Owning Team            | owningteam                 | Lookup            | Standard | ✓            | Optional |            |

## User Interface (UI) and User Experience (UX)

When building a Model-driven app, most of the UI and UX are predetermined for you. You define the data model to build from, and then Power Apps determines the controls used in the app. You can influence these controls by determining what table assets you include. You define in the App Designer What Forms, Views, Charts, and Dashboards are used in the app. You also control the navigation options via the Site Map. As you are planning your app, determine which components are needed in the app design, and create them before building your app.

To continue building off of the example we've been using throughout this module, below is a simple Model-driven Form, which captures various pieces of information for creating a New Fundraiser.

Here is an example of what the New Fundraiser form looks like when editing from the App Designer.



## Business Logic

When incorporating business logic in your app, there are two primary options available. You can set **Business Rules** on your Dataverse tables or you can build **Business Process Flows**.

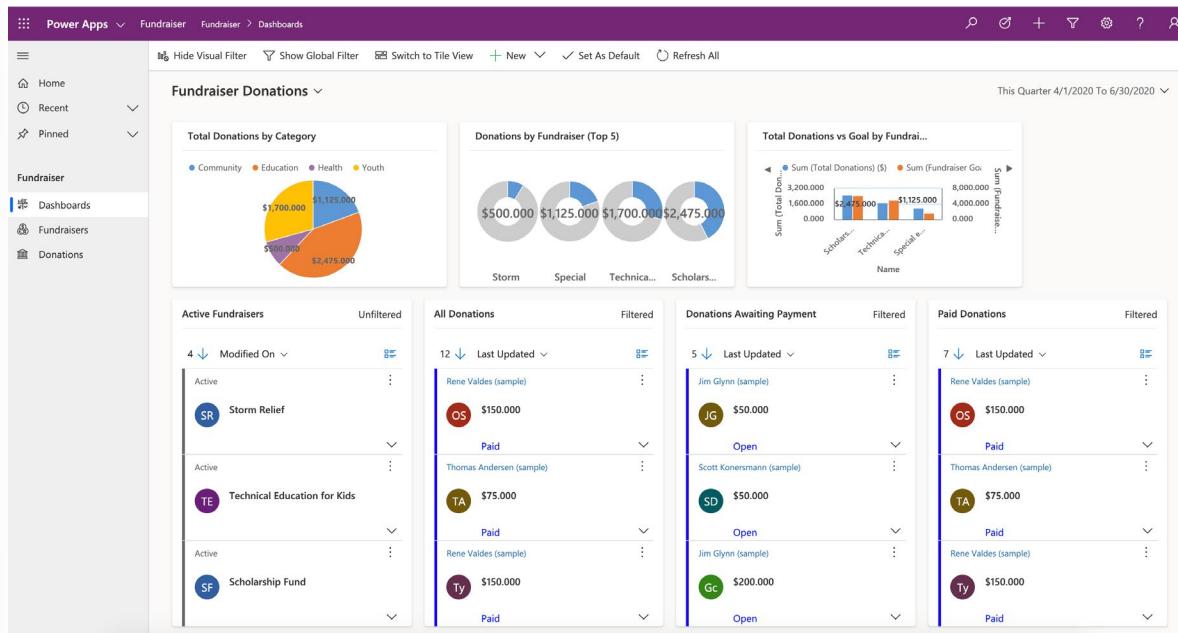
With **Business Rules**, you will define behaviors at the data layer. This is great for changing when a column is required, setting a default value, or even showing or hiding a column based on a criteria. An example could be a table for tracking expenses. You could have a column for type of travel and then build a business rule that says if they choose automobile then the mileage column is required, else it is optional. This gives you great power to make sure you maintain data consistency in all scenarios.

**Business Process Flows** are used to guide users through using your app. These workflows can provide visuals on next steps based on the status of the data and facilitate other actions that you want to occur as the user uses the app. Business Process Flows let you bring automation to your app and make it more of a guided experience than just a place to enter data.

## Output

A common output need for apps is to visualize the data. For this requirement, you can implement Dashboards with custom filters and visual graphics to tie all of this data together right in your app. When creating your Dashboards, make sure it's simple for your users to consume without overwhelming them

with all the data. Provide high-level snapshots of your data and allow them to use filters to dive deeper into the data if needed.



## Additional third-party solutions and app accelerators

It is also important to know about the different App accelerators and third-party solutions available to you. Depending on the industry you are in, Health, Financial, Banking, Education, Non- Profit, Automotive, or Media, Microsoft has released a number of accelerators or foundational components to assist you with quickly standing up your solution. For more information, see [Industry accelerators overview<sup>12</sup>](#).

For more information. see [Planning a Power Apps project<sup>13</sup>](#).

## Exercise - Create a model-driven app

In this unit, you'll create a model-driven app by using one of the standard tables that's available in your Microsoft Power Apps environment.

### Create a model-driven app

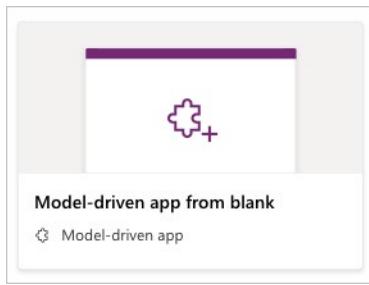
1. Sign in to [Power Apps<sup>14</sup>](#) by using your organizational account.
2. Select the environment you want, or go to the [Power Apps admin center<sup>15</sup>](#) to create a new one.
3. On the **Home** page, select the **Model-driven app from blank**.
4. Click **Create**.

<sup>12</sup> <https://docs.microsoft.com/common-data-model/industry-accelerators?azureportal=true>

<sup>13</sup> <https://docs.microsoft.com/powerapps/guidance/planning/introduction>

<sup>14</sup> <https://make.powerapps.com/>

<sup>15</sup> <https://admin.powerapps.com/>

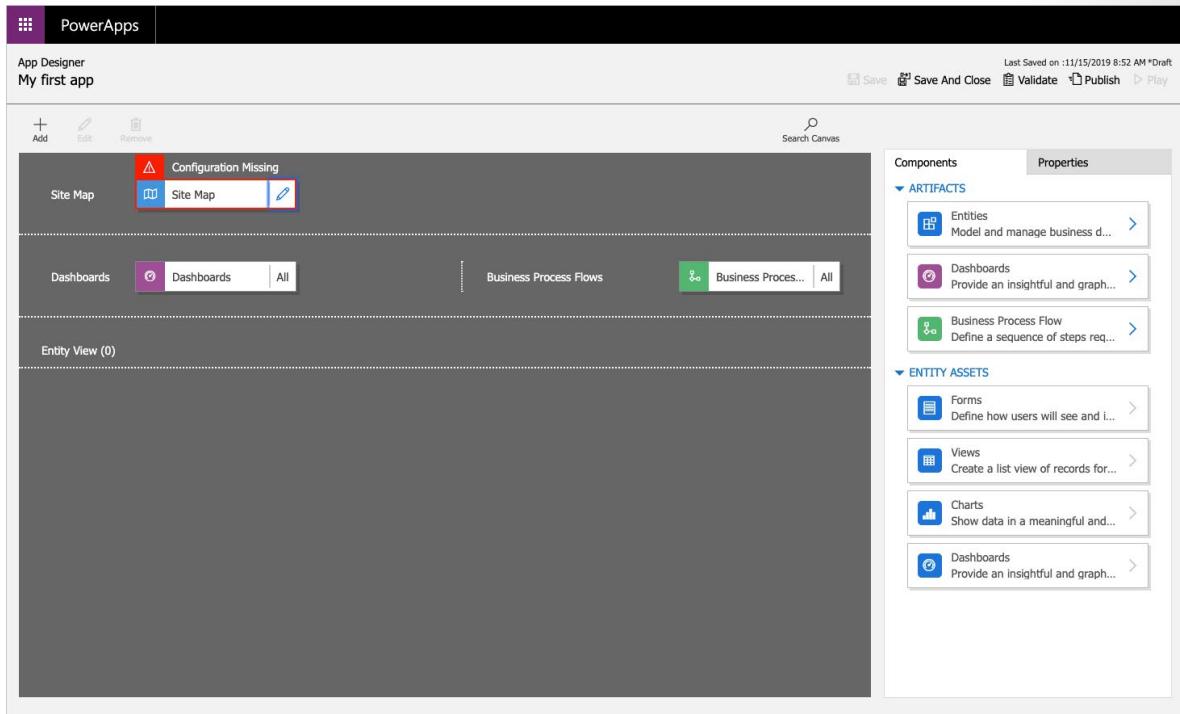


5. On the **Create a New App** page, enter a name and description for the app.
6. Select **Done**. Your new app appears in the App Designer, and you can now add components to it.

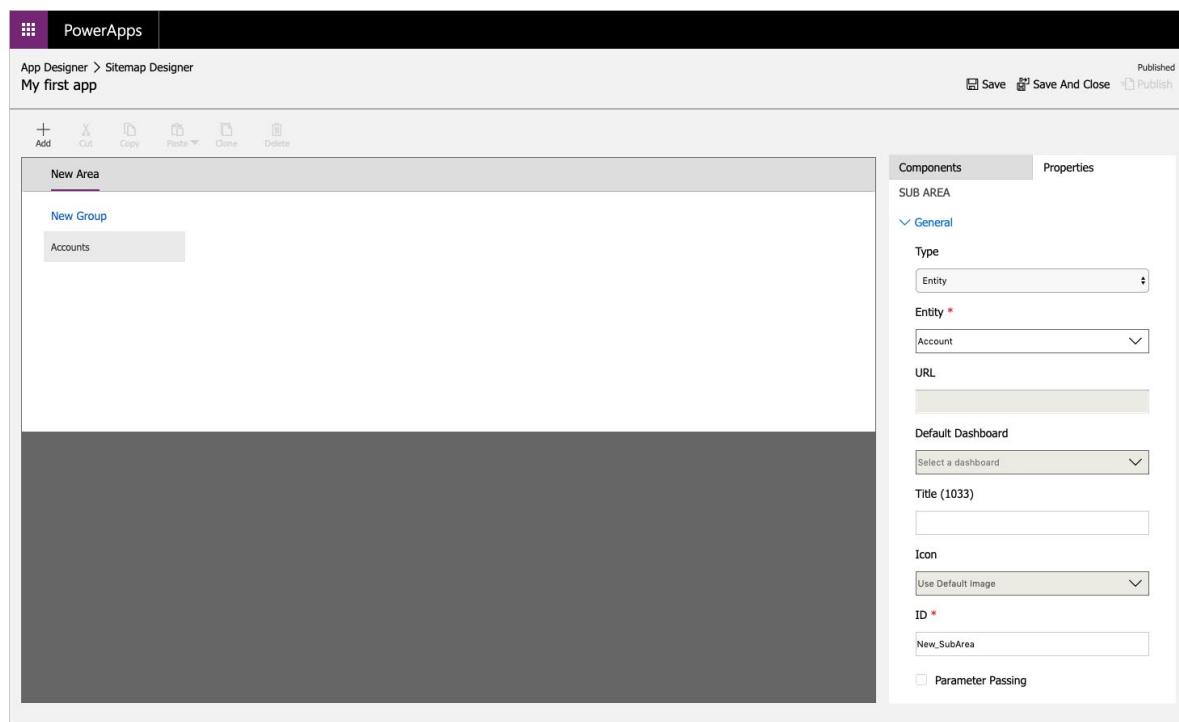
## Add components to your app

You add components to your app by using the App Designer.

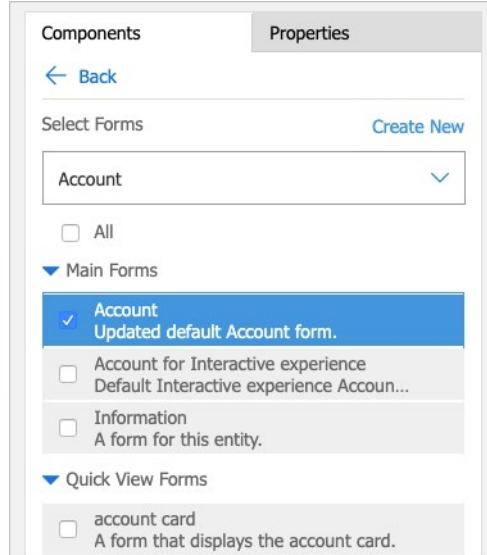
1. Select the **Open the Site Map Designer** pencil icon to open the site map designer.



2. In the site map designer, select **New Subarea**, and then, in the right pane on the **Properties** tab, select the following properties:
  - **Type:** *table*
  - **table:** *Account*



3. Select **Save And Close**.
4. In the App Designer, select **Forms**, and then, in the right pane under **Main Forms**, select the **Account** form.



5. In the App Designer, select **Views**, then select the following properties:
  - Active Accounts
  - All Accounts
  - My Active Accounts
6. In the App Designer, select **Charts**, then select the **Accounts by Industry** chart.

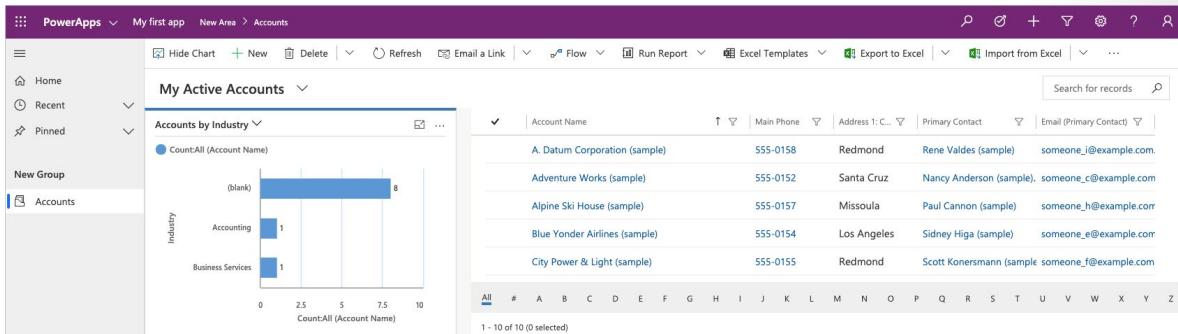
- On the App Designer toolbar, select **Save**.

## Publish your app

On the App Designer toolbar, select **Publish**. After you publish the app, it's ready for you to run or share with others.

Above My Active Accounts, select **Show Chart**.

If the sample data for your accounts does not have an Industry populated, go into a few accounts and add an Industry. Once you have updated a few accounts with an industry the chart will update as well.



## Incorporate business process flows

You can help ensure that people enter data consistently and follow the same steps every time they work with a customer by creating a business process flow.

For example, you might want to create a business process flow to have everyone handle customer service requests the same way, or to require that people get approval for an invoice before submitting an order. Business process flows use the same underlying technology as other processes, but the capabilities that they provide are very different from other features that use processes. To learn how to create or edit a business process flow, see [Create a business process flow](#)<sup>16</sup>

## Use business process flows

Why should you use business process flows? Business process flows provide a guide for people to get work done. They provide a streamlined user experience that leads people through the processes their organization has defined for interactions that need to be advanced to a conclusion of some kind. This user experience can be tailored so that people with different security roles can have an experience that best suits the work they do.

You should use business process flows to define a set of steps for people to follow to take them to a desired outcome. These steps provide a visual indicator that tells people where they are in the business process.

Business process flows reduce the need for training because new users don't have to focus on which table they should be using. They can let the process guide them.

<sup>16</sup> <https://docs.microsoft.com/power-automate/create-business-process-flow>

You can configure business process flows to support common sales methodologies that can help your sales groups achieve better results.

For service groups, business process flows can help new staff get up to speed more quickly and avoid mistakes that could result in unsatisfied customers.

## System business process flows

The following business process flows can be found in Power Automate. To understand how business process flows work, review these system business process flows:

- Lead to Opportunity Sales Process
- Opportunity Sales Process
- Phone to Case Process

## Multiple tables in business process flows

You can use a business process flow for a single table or to span multiple tables. For example, you may have a process that begins with an opportunity, then continues to a quote, an order, and then an invoice, before finally returning to close the opportunity.

You can design business process flows that tie together the rows for up to five different tables into a single process so that people using the app can focus on the flow of their process rather than on which table they are working in. This way, they can easily navigate between related table rows.

## Multiple business process flows are available per table

Not every user in an organization may follow the same process and different conditions may require that a different process be applied. You can have up to 10 active business process flows per table to provide appropriate processes for different situations.

## Scenario

In the exercise in the previous module of this learning path, you created the Prospects table in Dataverse and imported the existing leads, now in this exercise you will use this data to create a model-driven app. This app will allow the sales team to enter and edit leads on the go and keep the managers up to date on the current leads and forecasted revenue.

## Create the model-driven app for the prospects table

1. Sign in to **Power Apps**<sup>17</sup> by using your organizational account.
2. Select the environment you want, or go to the **Power Apps admin center**<sup>18</sup> to create a new one.
3. On the **Home** page, select the **Model-driven app from blank**.
4. Select **Create**.

---

<sup>17</sup> <https://make.powerapps.com/?azure-portal=true>

<sup>18</sup> <https://admin.powerapps.com/?azure-portal=true>

5. Enter the following and Select **done**.
  - Name: Prospect Entry
6. Select the **Open the Site Map Designer** pencil icon to open the site map designer.
7. Select **New Subarea**.
8. On the right, for **Title** enter **Prospects Area**.
9. Click the drop down for **select a type**, then choose **table (table)**.
10. For table (table), select **Prospects**.
11. Select **New Group**.
12. On the right, for **Title** enter **Prospects Group**.
13. Click **Save**.
14. Click **Publish**.

## Creating a chart

1. Select **Charts**.
2. Select **Create New**.
3. For the Chart Name, enter **Forecasted Revenue by Stage**.
4. For Legend Entries (Series), check the box, and select the **Forecasted Revenue** column.
5. For Horizontal (Category) Axis Labels, select the **Stage** column.
6. Select **Save and Close**.
7. Back on the App Designer, select the checkbox next to **Forecasted Revenue by Stage**.
8. Select **Save**.
9. Select **Publish**.

## Creating the form

You can attempt to create a new form in the App Designer but if you run into any issues, you can follow the steps below to create a new main form.

1. Go to the Power Apps Home Page, and on the left, Select **Data**.
2. Select **tables**.
3. Locate and select the **Prospects** table.
4. Select **Forms**.
5. Select the drop-down arrow next to Add form, and then select **Main Form** from the drop down. A new window will open.
6. Drag the **Stage** column from the right and place it below the **Owner** column in the center.
7. Drag the **Contract Amount** column and place it below the **Stage** column.
8. Drag the **Probability** column and place it below the **Contract Amount** column.

9. Drag the **Forecasted Revenue** column and place it below the **Probability** column.
10. Now select **Forecasted Revenue** and then select **Change Properties** on the ribbon.
11. Select the checkbox for **Read-only column**.
12. Select **Save**.
13. Select **Publish**.
14. Close the window.
15. Select **Done**.

## Summary

Congratulations on creating your first model-driven app!

Defining and enforcing consistent business processes is a key aspect of model-driven app design. Consistent processes help ensure that your app users can focus on their work and not have to remember to perform a set of manual steps.

Let's review what you've learned:

- Model-driven app design is an approach that focuses on quickly adding components to apps. These components include dashboards, forms, views, and charts.
- Little or no code is required to build model-driven apps.
- Model-driven design uses metadata-driven architecture so that designers can customize their apps.
- The best way to get started building model-driven apps is to use sample apps and data; then customize the apps.

# Create and manage columns within a table in Dateverse

## Define columns in Dateverse

columns are a way to store a discrete piece of information within a record in a table. columns have types, meaning that you can store data of a certain type in a field that matches that data type. For example, if you have a solution that requires dates, then you would store the date in a field with the type of Date. Similarly, if you want to store a number, then you store the number in a field with the type of Number.

The number of columns within a table varies from a few columns to a hundred or more. If you need more than a few hundred columns in a table, you might want to reconsider how you are structuring data storage for your solution because, likely, there is a better way.

Every database in Dataverse starts with a standard set of tables and each standard table has a standard set of columns.

**Tip:** Always use standard tables and columns when possible. You can rename a table if that makes the table more understandable in the context of your solution. Always review the list of standard tables and make sure a standard table will not meet your needs before you create a new table.

## Column types in Dataverse

The column type determines the values that can be stored within that column. All columns have one and only one column type. The column types that are available in Dataverse include:

**Text** - A text value intended to be displayed in a single-line text box.

**Text Area** - A text value intended to be displayed in a multi-line text box. If you require more than 4,000 characters, use a Multiline Text data type.

**Email** - A text value that is validated as an email address and rendered as a mailto link in the column.

**URL** - A text value that is validated as a URL and rendered as a link to open the URL.

**Ticker Symbol** - A text value for a ticker symbol that will display a link that will open to show a quote for the stock ticker symbol.

**Phone** - A text value that is validated as a phone number rendered as a link to initiate a phone call by using Skype.

**Whole Number** - A number value presented in a text box.

**Duration** - A number value presented as a drop-down list that contains time intervals. A user can select a value from the list or type an integer value that represents the number of minutes.

**Timezone** - A number value presented as a drop-down list that contains a list of time zones.

**Date and Time** - A date and time value.

**Date Only** - A date and time value that only displays a date. The time value is stored as 12:00 AM (00:00:00) in the system.

**Language** - A number value presented as a drop-down list that contains a list of languages that have been enabled for the environment. If no other languages have been enabled, the base language will be the only option. The saved value is the Locale Identifier (LCID) value for the language.

**Currency** - A money value for any currencies that are configured for the environment. You can set a level of precision, or you can choose to base the precision on a specific currency or a single standard precision that is used by the organization.

**Decimal Number** - A decimal value with up to 10 points of precision. See [Using the right type of number<sup>19</sup>](#) for more information.

**Floating Point Number** - A floating point number with up to 5 points of precision. See [Using the right type of number<sup>20</sup>](#) for more information.

**Image** - Displays a single image for each record in the application. Each table can have one image column. The name that you enter when creating an image column will be ignored. Image columns are always named **tableImage**. You can only have one image column for each table.

**Lookup** - Creates a reference to a single row for a single target row type.

**Choices** - Displays a list of options where more than one option can be selected.

**Multiline Text** - A text value intended to be displayed in a multi-line text box. This value is limited to a maximum of 1,048,576 characters. You can also set a lower Max Length.

**Choice** - Displays a list of options where only one can be selected.

**Yes/No** - Displays two options where only one can be selected. You can choose which labels are displayed for each option. The default values are Yes and No.

Tip: You can add any combination of columns to a custom or standard table to meet your needs, but you can't delete a standard column from a standard table.

## Add a column to a table

You can add columns when you create a new custom table, or you can add columns to an existing table at any time. Adding a new column is the same whether you are creating a new table or adding to an existing table.

---

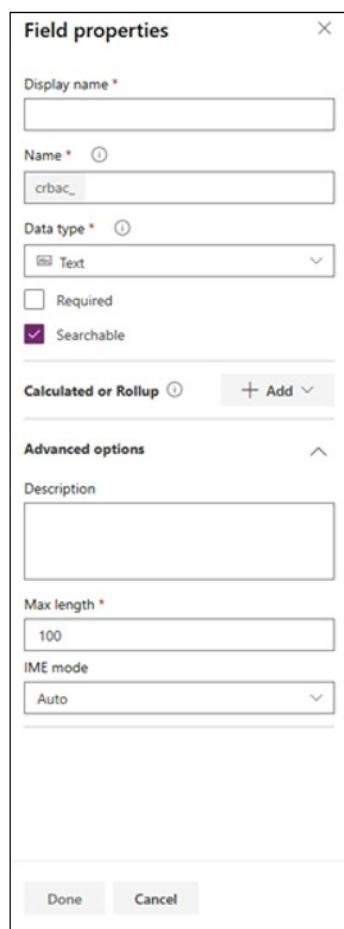
<sup>19</sup> <https://docs.microsoft.com/powerapps/maker/common-data-service/types-of-fields#using-the-right-type-of-number>

<sup>20</sup> <https://docs.microsoft.com/powerapps/maker/common-data-service/types-of-fields#using-the-right-type-of-number>

**TIP:** Before you create a custom column, evaluate whether an existing column meets your requirement.

1. Sign in to Power Apps.
2. Select the **Data** option on the left-hand menu.
3. Open an existing table or create a new table.
4. Select **Add column** on the menu at the top of the page.
5. Enter information in the following columns:
  - a. **Display name** - The name that is shown to users.
  - b. **Name** - This is the internal name that is used by your application.
  - c. **Data type** - This is the type of data that you want to store in the column.
  - d. **Required** - Select this check box if you want to ensure that this column always has a value when a user tries to add a record to this table.
  - e. **Searchable** - Clear this check box for columns for the table that you do not use. When a column is searchable, it appears in **Advanced Find** and is available when you are customizing views. Clearing this check box reduces the number of options that are shown to people who are using **Advanced Find**, and helps make it easier for users to create custom views without seeing unused columns. Clear the **Searchable** check box when you are working with columns in a standard or complex table that you do not use.
  - f. **Calculated or Rollup** - Use to define a calculation or a rollup in this column.
  - g. **Description** - This is a description of the column.
  - h. **Max Length** - Use to define the maximum length of the data that a user can enter in this column. Note that this is used with text columns.
  - i. **Minimum and Maximum Values** - This column is available and used with number columns.
  - i. **Minimum and Maximum Values** - This column is available and used with number columns.

- i. **Minimum and Maximum Values** - This column is available and used with number columns.



## Create a primary name column

A **Primary Name** column is always created by Dataverse when you create a new custom table. This is the first column that is listed and available so you have a way to identify a record in the new custom table by using a business value or an autogenerated whole number.

The **Primary Name** column is not the same as the internal key column that is also autogenerated when you create a new custom table. The **Primary Key** column is a GUID. You can change the name of the **Primary Name** column to make it friendlier for business users. You can also choose between a text column or autogenerated whole number.

*Tip:* If you want to make the primary column unique, then make a key and assign the **Primary Name** column to the new key.

Note: If you have duplicated data in the **Primary Name** column in multiple rows, the key will not be created. You can only create a key if data in the column's existing row is unique across all rows in the table.

## Restrictions that apply to columns in a table

When using Dataverse for your business solutions, you should keep a few restrictions in mind.

Maximum Number of columns in a table:

- There is no hard upper limit on the number of columns that you can have in a table, but there is an upper boundary due to limits in how much data you can store in a single record. It is difficult to provide a specific number because each type of column can use a different amount of space. The upper limit depends on the total space that is used by all the columns for the table.
- As a rule, you should have less than a few hundred columns in a table. If you have more than a few hundred columns in a table, then you should look at restructuring how you have designed the tables in your solutions and try to split the table with an excessive number of columns into more than one table.

**Rollup** columns:

- Max of 100 for each organization
- Max of 10 for each table
- Can't trigger workflows

**Choices** columns:

- Choice columns provide a set of options that will be displayed in a drop-down control on a form or in picklist control when you are using Advanced Find. Dynamics 365 can support thousands of options within a Choice, but you should not consider this as the upper limit. Usability studies have shown that people have trouble using a system where a drop-down control provides a large number of options.
- Use a Choice column to define categories for data. Don't use Choice columns to select categories that actually represent separate items of data. For example, rather than maintaining a Choice column that stores each of hundreds of possible manufacturers of a type of equipment, consider creating a table that stores references to each manufacturer and use a lookup column instead of a Choice column.

**Primary Name** column

- You can only have one primary column for each custom table.

**Searchable** columns

- You can only have five searchable columns for each table.

## Create an auto numbering column

Autonumber columns automatically generate alphanumeric strings whenever they are created. You can customize the format of these columns and then rely on the system to generate matching values that automatically fill them in.

Autonumber columns appear in the Text category of columns when you create a new column. You can also activate autonumber functionality in an

existing text column by opening the column and selecting **Autonumber** in the datatype dropdown. Similarly, auto number functionality can also be disabled at any time by opening the column and selecting text as the type in the **Data type** drop-down list.

Autonumber columns offer many options for the type of auto-generated number that you want to generate. You can create any of the following autonumber type columns.

**String prefixed number** - The most common auto number format is a simple string prefixed number. When this type is selected, the autonumber will consist of an automatically incrementing number with an optional string constant prefix. For example, a string prefixed number with the prefix of Contoso would generate rows such as Contoso-1000, Contoso-1001, Contoso-1002, and so on.

**Date prefixed number** - Another common auto number format is a date prefixed number. When this type is selected, the auto number will consist of an automatically incrementing number with a formatted date prefix. The date portion of the row will reflect the current date and time at which the row was created in UTC time. For example, a date prefixed number would generate rows such as 2019-26-02-1000, 2019-27-02-1000, 2019-28-02-1000, and so on, depending on the current date and selected date format.

**Custom** - For more advanced makers with specific use cases, you have the option to fully customize the desired format of an autonumber column. The format might consist of string constants, automatically incrementing numbers, formatted dates, or random alphanumeric sequences. For detailed information about how to define custom formats, see **AutoNumberFormat options<sup>21</sup>**.

## Seeding a starting value

You can also set a *seed* value so you can start autonumbering at any value. The seed value of an autonumber column is the starting number that is used for the number portion of the format. For example, if you want an autonumber column to generate rows such as Contoso-1000, Contoso-1001, Contoso-1002, and so on, then the desired seed value is 1000 because that is the value that your number sequence starts with. Autonumber columns have a default seed value of 1000, but you can set a custom seed value if you want.

## Create an alternate key

It is common to need a way to uniquely identify a record in a table. By default, Dataverse tables have a GUID as their only unique column. This GUID is called the **Primary Key** and it consists of a long string of numbers and letters that are not useful to a regular user regarding meaning or significance.

Defining a new key for a table allows you to identify a record in a more meaningful way by using a column that is familiar to users. When you define a column as a key, Common Data Model makes sure that

---

<sup>21</sup> <https://docs.microsoft.com/dynamics365/customer-engagement/developer/create-auto-number-attributes#autonumberformat-options>

every entry in that key column is required and unique so you can use the key column to distinctively identify a specific record. This can be especially helpful if you are integrating your data with an external system that uses that ID or number to identify a record (and not the Dataverse GUID). It also improves the search and filtering on the particular column because alternate key fields are always indexed.

Finally, keys can be based on a single column (Order ID) or a combination of fields, such as Financial year and Order ID. If you set a key on a column and try to enter duplicate data, the record will fail to save.

Note: You are limited to five alternate keys for each table.

Set up an alternate key for an table with the following steps:

1. Sign in to Power Apps and select the **Data** menu option in the left-hand column.
2. Select the table that you want to add a new key to.
3. Select **Keys** from the options above the list of fields.
4. Select **Add key** in the upper left of the screen.
5. Select one or more fields that will make up the new key.
6. Give the key a name.
7. Select the **Done** button.

It will take a few minutes for Dataverse to create the new key and indexes. Then, you can start using it in your business solution.

Tip: If you have duplicated data in a column that is used by the key in multiple rows, then the key will not be created. You can only create a key if the data in the column's existing row is unique across all rows in the table.

## Add columns to a custom table

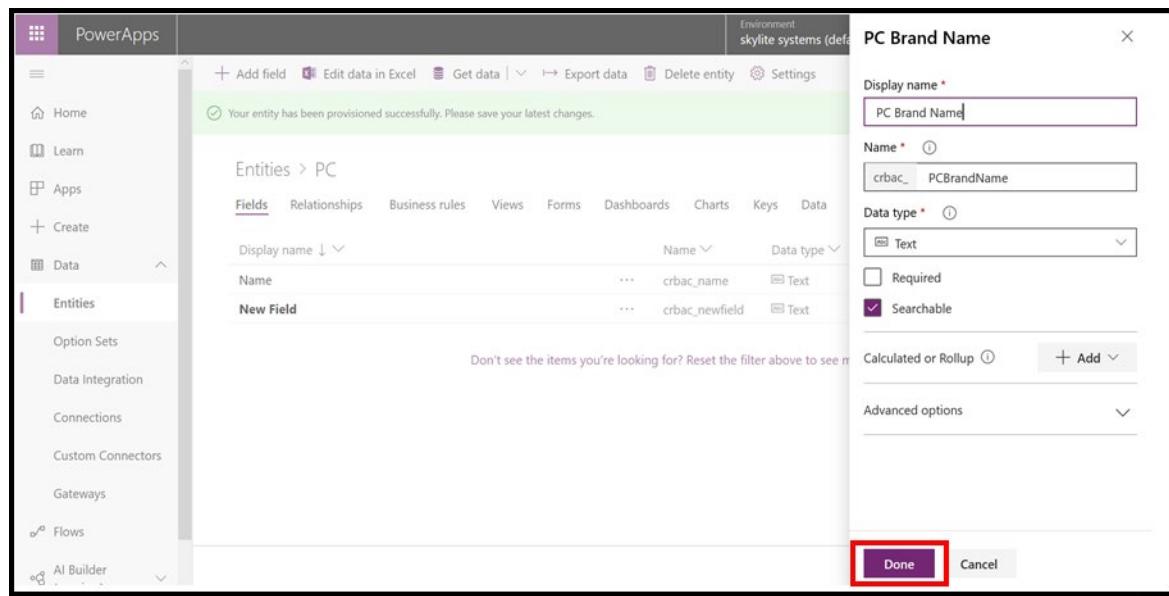
In this lab, you will add a few columns to the custom table PC Manufacturers.

If you have not created the custom table as a part of this learning path, you can complete the steps [here<sup>22</sup>](#) to complete these exercises.

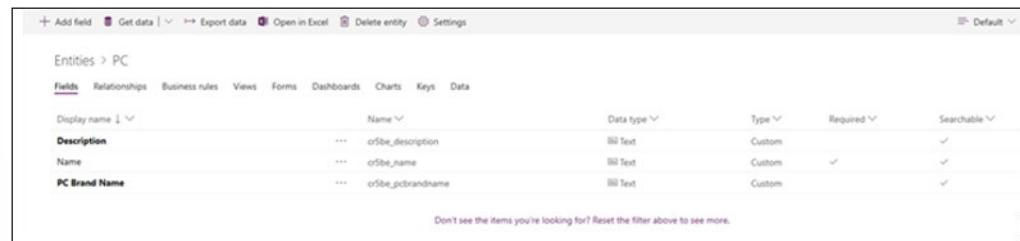
1. Sign in to Power Apps.
2. Select the environment where you added the new custom table PC Manufacturers.
3. Select **Data** on the panel on the left-hand side of the Power Apps Portal to expand the available choices.
4. Select **tables** under **Data** on the left-hand side of the portal.
5. Select the new custom table **PC** in the list of tables.  
This will open the list of columns in the table.

<sup>22</sup> <https://docs.microsoft.com/learn/modules/create-manage-entities/5-exercise>

6. Select **+ Add column** on the menu above the list of columns in the table.
7. Type **PC Brand Name** in the **Display name** column.
8. Keep the **Searchable** check box selected.
9. Select **Done**.



10. Select **+ Add column** on the menu above the list of columns in the table.
11. Type **Description** in the **Display name** column.
12. Select **Text area** in the Data type.
13. Select the **Done** button.
14. Select the **Save table** button to save changes.



After you select the **Save table** button, the custom PC Manufacturers table should have the newly added columns, as shown in the previous figure.

## Rename a primary column

In this lab, you will rename the primary column that was created by default when you created the custom table PC Manufacturers.

1. Sign in to Power Apps.

2. Select the environment where you added the new custom table PC Manufacturers.
3. Select **Data** in the panel on the left-hand side of the Power Apps Portal to expand the available choices.
4. Select **tables** under the **Data** option on the left-hand side of the portal.
5. Select the new custom table **PC Manufacturers** in the list of tables. This will open the list of columns in the table.
6. Recall that the **Name** column was autocreated when you created the custom table. Notice that it is selected as both **Searchable** and **Required**, as shown in the following figure.

| Display name  | Name              | Type | Required | Searchable                          |
|---------------|-------------------|------|----------|-------------------------------------|
| Description   | cr5be_description | Text | Custom   | <input checked="" type="checkbox"/> |
| Name          | cr5be_name        | Text | Custom   | <input checked="" type="checkbox"/> |
| PC Brand Name | cr5be_pcbrandname | Text | Custom   | <input checked="" type="checkbox"/> |

7. Select the **Name** column and update the *Display Name* field to **PC Model Name**, as shown in the following figure. You cannot change the internal name (name\*) of the column.

PC Model Name

Display name \*

PC Model Name

Name \*

cr5be\_name

Data type \*

Text

Required

Searchable

Advanced options

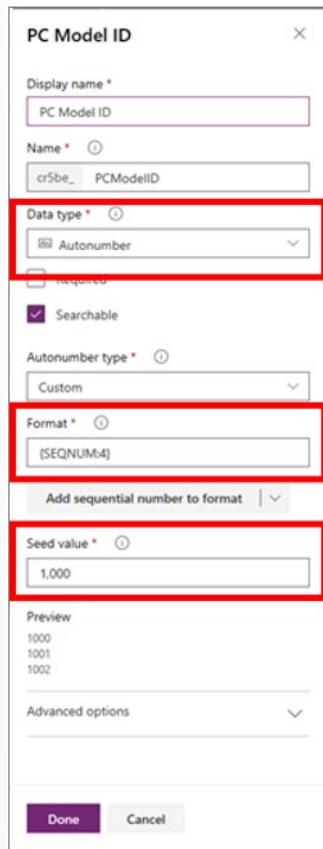
Done Cancel

8. Select **Done**.
9. Select **Save table**.

## Add an autonumber column

In this lab, you will add an autonumber column to help identify the manufacturer record in the custom table.

1. Sign in to the Power Apps portal.
2. On the left pane, expand **Data** and select **tables**.
3. Select the custom table **PC Manufacturers**.
4. On the toolbar, select **Add column**.
5. On the right pane, enter a **PC Model ID** for the **Display name** and select **Autonumber** for the **Data type**.
6. Under **Autonumber type**, select **Custom**.
7. Leave the **Format** as the default of **{SEQNUM:4}**.
8. Customize a **Seed value** or keep the default value of **1,000**, as shown in the following figure.



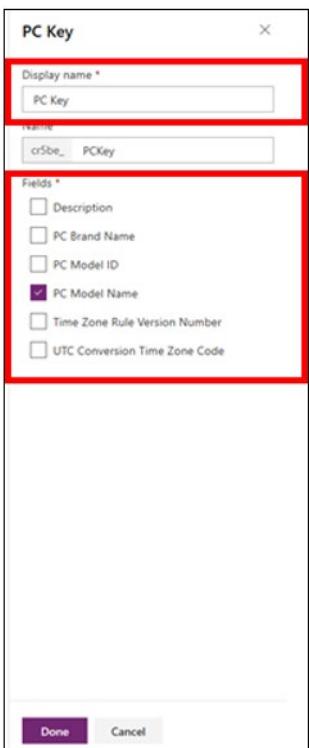
9. Select **Done**.
10. Select the **Save table** button.

## Create a key

Create a key to ensure that the value is unique and indexed.

1. Sign in to Power Apps.
2. Select the environment where you added the new custom table PC Manufacturers.
3. Select **Data** in the panel on the left-hand side of the Power Apps Portal to expand the available choices.
4. Select **tables** under the **Data** option on the left-hand side of the portal.
5. Select the new custom table **PC Manufacturers** in the list of tables. This will open the list of columns in the table.
6. Select **Keys** on the menu at the top of the list of columns in your custom table PC Manufacturers.
7. Select **Add Key** on the menu at the top of the list of keys.
8. Enter **PC Key** in the **Display name** column.
9. Select the check box next to the **PC Model Name** column.

Note: A key can be made up of one or many columns. If you select multiple columns, then the key is called a compound key.



10. Select the **Done** button.
11. Select the **Save table** button.

After selecting **Done**, you will see the new key added to the list of all keys that are created for this table. You can create up to five keys for each table. All values in the key will be unique.

## Summary

In this module, you learned about:

- The columns in Dataverse.
- Types of columns that are available in Dataverse.
- Adding a column to an table.
- The **Primary Name** column and what it's used for.
- Restrictions that are associated with columns in Dataverse.
- Creating an autonumbering column.
- Creating an alternate key.

# Get started with security roles in Dataverse

## Introduction to environment roles

Each environment has zero or one instance of a Microsoft Dataverse database, and your organization can have many environments that are available to many different groups of users at a time. It is common practice to set up an environment so you can limit who can access the data, apps, and workflows within it.

Classic software lifecycle management provides a good use case of why you might want to set up different environments in Dataverse.

Consider the following example. It is common to segment environments for development, test, and production. Dataverse allows you to set up a development environment and limit access so only developers and a few managers or test users have permission to access the data and apps in that development environment. You can then set up a test environment and set up permissions so that a few test users and developers have access to it and the data within the instance of Dataverse within that environment. Finally, you can set up production environment permissions so that a wide audience of users has access to the production environment and the data in the instance of a Dataverse database, Power Apps, and Power Automates within that production environment.

*Important:* Access to an environment does not give a user access to any data, apps, or workflow within that environment. Users must be given explicit access to data by an administrator in Dataverse while the maker who creates an app, connector, or workflow must grant access to their work products.

## Understand environment roles

You can manage environment security by using roles and then adding users to the environment and assigning roles to users. A role has certain permissions that are associated with it, and you can associate a user with one or many roles. Environments have two built-in roles that provide access to permissions within an environment, and you'll assign users to one of these two roles when considering what permissions you want to give to a user in an environment.

The built-in environment roles are:

- System Administrator
- Environment Maker

This unit examines each role to help you understand how it works within an environment.

*Important:* A user is automatically associated with the Environment Maker role when they are added to an environment.

## System Administrator role

The System Administrator role can perform all administrative actions on an environment, including the following tasks:

- Add or remove a user or group from either the Environment Admin or Environment Maker role.

- Provision a Microsoft Dataverse database for the environment.
- View and manage all resources that are created within an environment.
- Set data loss prevention policies.

## Environment Maker role

The Environment Maker role can create resources within an environment such as apps, connections, custom connectors, gateways, apps that use Power Apps, and flows that use Power Automate. The following applies to members of the Environment Maker role:

- Environment Makers can distribute the apps that they build in an environment to other users within an organization by sharing the app with individual users, security groups, or to all users in the organization.
- Users or groups that are assigned to these environment roles are not automatically given access to the environment's database (if it exists) and must be given access separately by a Database owner.
- Whenever a new user signs up for Power Apps, they are automatically added to the Maker role of the default environment.
- When you add a user to an environment, they are assigned two roles by default.
  - Dataverse User (this role is created when you instantiate an instance of a Dataverse database and all users in the environment are assigned this role)
  - Environment Maker

Users or security groups can be assigned to either of these two roles by a System Administrator from the Power Apps Administration Center.

## Exercise-Adding or disabling an environment user

You can add users to any environment that you create in Microsoft Dataverse.

The following steps will help you add users from your tenant to an environment.

1. Sign in to Power Apps.
2. Select the gear icon in the ribbon and select the **Admin Center** option.
3. Select an environment.
4. Find the **Access** section on the right-hand side, and select **See all** under **Users**.
5. Select the **Add Users** button in the ribbon.
6. Add one or more users by entering their name or email address.
7. After you have selected the users you want to add, select **Add**.

If you want to disable a user within an environment, remove a license from the user or remove the user from the security group that's associated with an environment. Removing a user from the security group

doesn't remove the user's license. If you want to make the license available to another user, you have to remove the license from the user account that was disabled.

Removing a license, disabling a user, and removing a user from a security group is done with the **Microsoft 365 admin center**<sup>23</sup>. For more information, see **Disable a user account in an environment**<sup>24</sup>.

## Understand user security roles and security role defaults

Roles are groups of permissions that you can assign to a user to grant them access and various capabilities and functionality like read, delete, or edit of rows in a table within an environment. Roles are granular and can be assigned to one or many tables in an environment. Roles can also control certain actions like the ability to create a custom table or Choices. Additionally, users are associated with one or many roles, and associating a user with a role gives them access to data and functionality that is specified within that role.

User security roles are either:

- Standard and created with every instance of Dataverse.
- Custom and created by an administrator.

This unit examines each type of security role.

### Default user security roles

When you create a new instance of Dataverse in an environment, a database is created with standard tables and several default security roles are created. The following predefined roles are available every time you create a Dataverse environment by using the Power Apps portal. Unless otherwise noted, all the privileges have global scope.

Dataverse includes several default roles with different access levels to standard tables and actions. The default standard roles are listed in the following table.

<sup>23</sup> <https://admin.microsoft.com/>

<sup>24</sup> <https://docs.microsoft.com/power-platform/admin/create-users-assign-online-security-roles#disable-a-user-account-in-an-environment/?azureportal=true>

| Security role     | Database privileges*                                        | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------|-------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Environment Admin | Create, Read, Write, Delete, Customizations, Security Roles | <p>The Environment Admin role can perform all administrative actions on an environment, including the following:</p> <ul style="list-style-type: none"><li>• Add or remove a user from either the Environment Admin or Environment Maker role.</li><li>• Provision a Data-verse database for the environment. After a database is provisioned, the System Customizer role should also be assigned to an Environment Admin to give them access to the environment's data.</li><li>• View and manage all resources created within an environment.</li><li>• Set data loss prevention policies.</li></ul> |
| Environment Maker | Customizations                                              | <p>Can create new resources associated with an environment, including apps, connections, custom APIs, gateways, and flows using Microsoft Power Automate. However, this role doesn't have any privileges to access data within an environment. More information: <b>Environments overview</b> (<a href="https://docs.microsoft.com/power-platform/admin/environments-overview/?azure-portal=true">https://docs.microsoft.com/power-platform/admin/environments-overview/?azure-portal=true</a>)</p>                                                                                                    |

| Security role        | Database privileges*                                                    | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|----------------------|-------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| System Administrator | Create, Read, Write, Delete, Customizations, Security Roles             | Has full permission to customize or administer the environment, including creating, modifying, and assigning security roles. Can view all data in the environment. More information: <b>Privileges required for customization</b> ( <a href="https://docs.microsoft.com/dynamics365/customer-engagement/customize/privileges-required-customization/?azure-portal=true">https://docs.microsoft.com/dynamics365/customer-engagement/customize/privileges-required-customization/?azure-portal=true</a> ) |
| System Customizer    | Create (self), Read (self), Write (self), Delete (self), Customizations | Has full permission to customize the environment. However, users with this role can only view records for environment entities that they create. More information: <b>Privileges required for customization</b> ( <a href="https://docs.microsoft.com/dynamics365/customer-engagement/customize/privileges-required-customization/?azure-portal=true">https://docs.microsoft.com/dynamics365/customer-engagement/customize/privileges-required-customization/?azure-portal=true</a> )                   |
| Basic User           | Read (self), Create (self), Write (self), Delete (self)                 | Can run an app within the environment and perform common tasks for the records that they own. This only applies to non-custom entities.                                                                                                                                                                                                                                                                                                                                                                 |
| Delegate             | Act on behalf of another user                                           | Allows code to <i>impersonate</i> , or run as another user. Typically used with another security role to allow access to records. More information: <b>Impersonate another user</b> ( <a href="https://docs.microsoft.com/powerapps/developer/common-data-service/impersonate-another-user/?azure-portal=true">https://docs.microsoft.com/powerapps/developer/common-data-service/impersonate-another-user/?azure-portal=true</a> )                                                                     |
| Support User         | Read Customizations, Read Business Management settings                  | Has full Read permission to customization and business management settings to allow Support staff to troubleshoot environment configuration issues. Does not have access to core records.                                                                                                                                                                                                                                                                                                               |

\*The scope of these privileges is global, unless specified otherwise.

**NOTE**

- Environment Maker and Environment Admin are the only predefined roles for environments that have no Dataverse database.
- The Environment Maker role can create resources within an environment, including apps, connections, custom connectors, gateways, and flows using Power Automate. Environment makers can also distribute the apps they build in an environment to other users in your organization. They can share the app with individual users, security groups, or all users in the organization. More information: **Share an app in Power Apps<sup>25</sup>**
- For users who make apps that connect to the database and need to create or update entities and security roles, you need to assign the System Customizer role in addition to the Environment Maker role. This is necessary because the Environment Maker role doesn't have privileges on the environment's data.
- If the environment has a Dataverse database, a user must be assigned the System Administrator role instead of the Environment Admin role for full admin privileges, as described in the preceding table.

**TIP**

Add the System Customizer role to a user if you want them to be able to create new entities.

When you add a user to an environment in Dataverse, the user is automatically assigned to the following:

- Security user roles - Basic User
- Environment roles - Environment Maker

## Exercise-Create a custom role

Dataverse has many standard default roles, but there might be times when you want to define a custom security role.

Dataverse supports the following eight different row-level privileges that can be used to define how a user interacts with data for one or more tables for use in building a custom role. The available row-level privileges for custom roles include:

**Create** - Required to make a new row. The rows that can be created depends on the access level of the permission that is defined in your security role.

**Read** - Required to open a row to view the contents. The rows that can be read depends on the access level of the permission that is defined in your security role.

**Write** - Required to make changes to a row. The rows that can be changed depends on the access level of the permission that is defined in your security role.

**Delete** - Required to permanently remove a row. The rows that can be deleted depends on the access level of the permission that is defined in your security role.

**Append** - Required to associate a row with the current row. For example, if a user has Append rights on an opportunity, the user can

---

<sup>25</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/share-app/?azure-portal=true>

add a note to an opportunity. The rows that can be appended depends on the access level of the permission that is defined in your security role.

**Append To** - Required to associate the current row with another row. For example, a note can be attached to an opportunity if the user has Append To rights on the note. The rows that can be appended to depends on the access level of the permission that is defined in your security role.

**Assign** - Required to give ownership of a row to another user. The rows that can be assigned depends on the access level of the permission that is defined in your security role.

**Share** - Required to give another user access to a row while keeping your own access. The rows that can be shared depends on the access level of the permission that is defined in your security role.

These row-level privileges can be grouped as needed and associated with a custom role. That custom role can then be applied to one or many tables as needed.

**TIP:** Roles can be copied so you can quickly create similar roles that might be slightly different.

## Create a custom security role and assign to tables and users

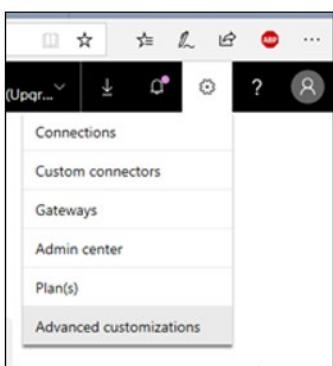
This lab will show you how to create a new role and associate that role with a custom table. Then, you can associate users to the new role so they can access the data in the custom tables as needed.

To grant access, you will need to do the following:

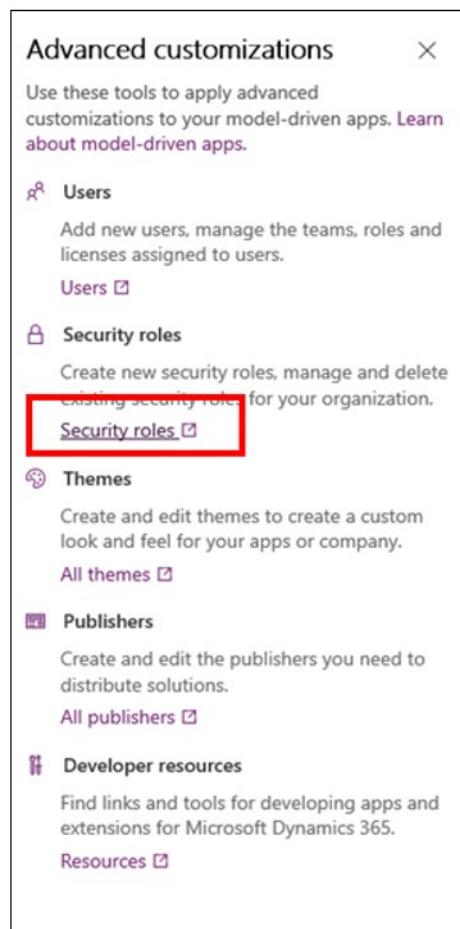
- Create a new user security role or amend an existing user security role to include settings for the custom table.
- Assign users to the security role.

To get started, use the following steps to create a new security role.

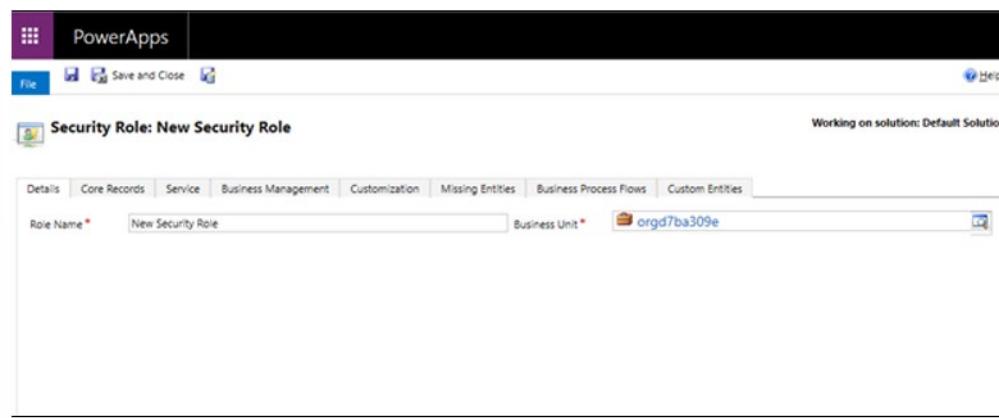
1. Sign in to Power Apps as an administrator.
2. Select the gear icon in the menu and select **Advanced Customizations**.



3. Select **Security Roles**. Note that this step can take a little time, depending on the number of instances in your tenant.



4. Select **New** in the menu bar, which will open the security role designer.
5. Enter a name for your security role in the **Role Name** column.



6. Locate the tables that your app uses by selecting each tab in the security role designer. If your tables are custom, they will be under the **Custom tables** tab.
7. When you have located your tables, select the privileges that you want to grant your users, such as Read, Write, Delete, and so on. Select the scope for performing that action by selecting the name of the table. Scope

determines how deep or high within the environment's hierarchy that the user can perform a particular action.

| Entity                | Create                | Read                             | Write                            | Delete                           | Append                | Append To             | Assign                | Share                 |
|-----------------------|-----------------------|----------------------------------|----------------------------------|----------------------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| Account               | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Activity              | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Contact               | <input type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input checked="" type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Customer Relationship | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Data Import           | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Data Map              | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Document Location     | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Email Signature       | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Email Template        | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Feedback              | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Follow                | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Import Source File    | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |
| Interaction for Email | <input type="radio"/> | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/>            | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> | <input type="radio"/> |

Key:

- None Selected
- User
- Business Unit
- Parent: Child Business Units
- Organization

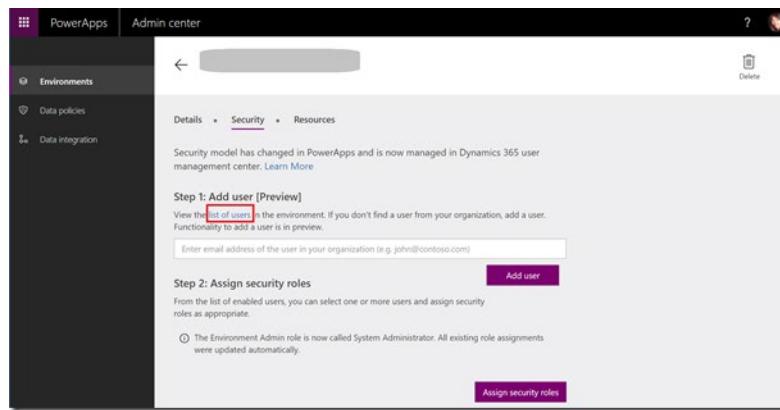
#### 8. Select **Save and Close**.

Congratulations, you have created a new custom security role. Next, you will assign users to this role.

To assign a user to a security role, you need to be a member of the System Administrator role in the current environment and then follow these steps:

1. Sign in to Power Apps as an admin, select the settings gear, and then select **Admin Center**.

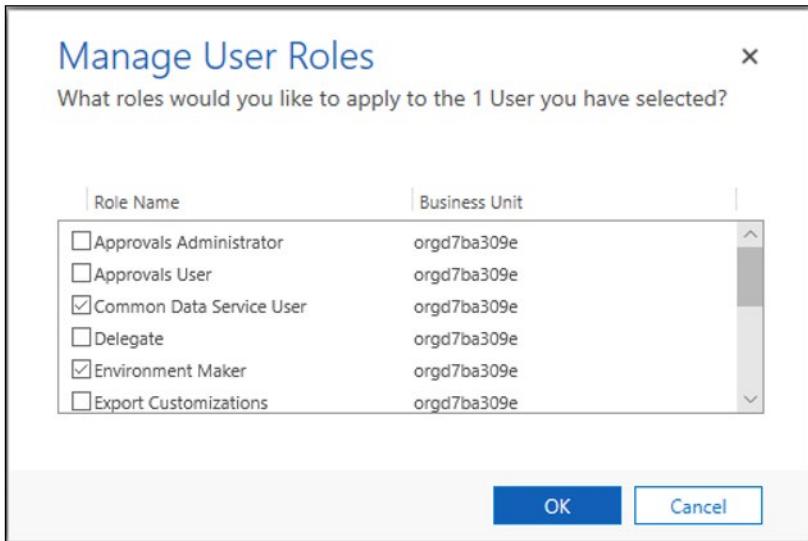
2. In Power Apps admin center, select the environment where you want to update a security role.
3. Select the **Security** tab.
4. Verify if the user(s) already exists in the environment by selecting **View the list of users** in the environment.  
If the user is not on the list, go to step 5.  
Otherwise, you can skip to step 6.



5. In case a user does not exist in the environment, you can add the user by entering the user's email address in your organization and then selecting the **Add user** button.
6. After you know the user(s) whom you want to assign a security role to exists in your environment, select the **Assign security roles** button.
7. Select the check boxes next to the user(s) that you want to assign and then select **Manage Roles**.

A screenshot of the Microsoft Dynamics 365 Admin center. The top navigation bar includes links for NEW, APPROVE EMAIL, REJECT EMAIL, PROMOTE TO ADMIN, MANAGE ROLES, CHANGE BUSINESS UNIT, CHANGE MANAGER, and CHANGE POSITION. Below this is a search bar labeled 'Search for records'. A section titled 'Enabled Users' shows a list of users with checkboxes next to their names. The users listed are: admin admin, Alan Steiner, Amy Adams, and Brad Sutton. All four users have checkboxes checked. The 'Business Unit' column shows 'orgd7ba309e' for all users. The 'Title' column shows 'System Administrator' for all users. The 'Position' and 'Main Phone' columns are empty.

8. In the **Manage User Roles** dialog box, in the **Role Name** section, select the check boxes next to the role(s) that you created in the previous section and make sure to also select the **Dataverse User** role (if it wasn't already). The Dataverse User role must be assigned to any user who wants to use your app or access Dataverse.

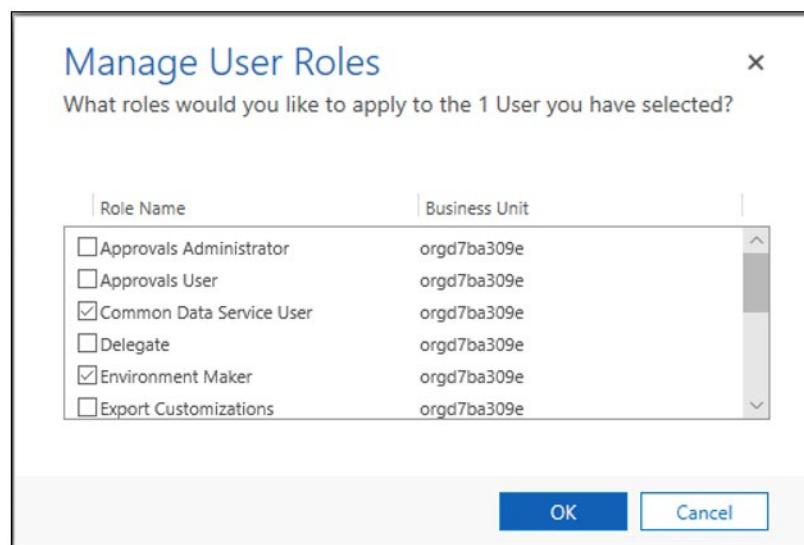


9. Select **OK** to assign the role(s) to the user that you selected.

## Exercise-Check the roles that a user belongs to

Checking the roles that a user belongs to is simple and you can do it from within Power Apps with the following steps:

1. Sign in to Power Apps as an admin.
2. Open the **Environments** option on the left-hand side of the page.
3. Select the environment where you want to check the user's permission settings.
4. Select the **Security** tab.
5. Select the **list of users** hyperlink under **Step One - Add User** option on the screen.
6. Select the check box next to the name of the user to view the security roles that the user has been assigned to.
7. In the **Manage User Roles** dialog box, select **Manage Roles** in the top menu to view user roles that are assigned to the user.



## Summary

In this module, you learned about security and the user in Common Data Service. In addition, you learned the following:

- What Environment Security roles are in Dataverse
- How to add or disable a user within an environment
- What User Security roles are, and which are assigned by default when you add a new user to an environment
- How to create a custom security role
- How you can check the roles that are assigned to a user

The following links were referenced within this module:

**Power Apps Admin Center<sup>26</sup>**

**Power Platform Admin Portal<sup>27</sup>**

**Power Apps<sup>28</sup>**

<sup>26</sup> <https://admin.powerapps.com>

<sup>27</sup> <https://admin.powerplatform.microsoft.com>

<sup>28</sup> <https://www.powerapps.com>

# Document and test your Power Apps application

## Create test plans

There are many types of testing that you should consider in your deployment planning. Each of these can be done either casually, or formally with a well-documented process to follow. Often in different deployments these will have slightly different expectations. Some testing options are manual (user adoption testing), and some are automated (load testing).

**Unit testing**- this is the smallest component of testing, a single unit of measure. Often this is thought to be a developer's job. However, every person on a project should include this in their work. An example of unit testing could be: You are given a requirement to add a calculated column to a form. Before you advance the task to the testing team, you would confirm the column was on the form as expected and test that the calculation was as expected.

**Quality assurance testing**- sometimes called QA, this is when someone independently tests the task by reviewing the expectation against the work performed. Often there will be testing of the smallest components, such as the units mentioned above and also testing of a full end-to-end user story.

**Performance testing** (sometimes also referred to as load testing)- there are many things that can be configured that will have an impact on system performance. Sometimes the number of concurrent users affects expected performance (load); sometimes it's the combined actions of these users, and sometimes an individual user could experience problems such as a slow-loading form that has too many subgrids. Integrations with other systems might also affect performance. Performance or load testing is most often done using external tools.

**Integration testing**- did the systems we integrated with perform as expected? Did we introduce unnecessary dependencies?

**User acceptance testing**- this is an opportunity to have direct feedback from the actual users of the system being built. User acceptance testing (hopefully) confirms the system built is the system users need.

Regardless of the exact type or plan you are using; the following considerations should be included.

## Use of Testing Plans

A testing plan is an important scoping document in your project plan to communicate what will and will not be tested and necessary considerations for success. The testing plan is designed to be carried out by a technical resource to catch bugs before user acceptance testing.

At a minimum, a test plan should provide a means to test each business requirement for the project. This gives the opportunity to show that each requirement was accommodated and accomplished. When there are issues in the future, a testing plan is a helpful reference of what was tested based on understandings at the time and what may be outside the original scope. The plan needs to define the level of detail to be tested, and the process for solving and retesting any issues found.

## Environment

Requirements of the test plan outline the environment needed to test accurately. The environment for testing should be as close to the technical setup of the production environment as possible. If the

application will be available in multiple formats such as desktop, mobile, and offline testing should be completed in all of these environments.

There may be times when the needed environment is not available at the time of testing. It is important that stakeholders are aware of these needs and dependencies and how they influence deadlines.

## Security

In creating a testing plan, it is important to consider all of the user personas that will use the application and test from the correct security roles. Common considerations on security include determining relevant security roles, column security profiles, business unit membership, and team membership. The testers mean of authenticating should be the same as the production users. For example, if users will access the application through a VPN, testing needs to be completed over VPN.

## Dependencies

Consider data dependencies such as child rows needed to populate lookups. If needed data is uncovered during testing be sure to note it so, you can remember to make the data available to users during user acceptance testing and the rollout to production.

Be mindful of customizations that reference specific rows such as a specific account or custom child row. If a customization such as a workflow or business rule relies on a specific row that row will need to be migrated, so the GUID is the same in both environments. Using the data import wizard will replicate the data in the new environment with a different GUID.

## Solution Awareness

Some configurations, such as Word and Excel templates, cannot be moved between environments and will need to be configured in each environment. Any configuration that must be complete in each environment should receive extra attention in testing.

## Resources

It is best practice to have components tested by a resource that did not create or configure the component.

## Test Plan Components

The following components are commonly found in testing plans.

- Environmental needs
- Features tested
- Features not tested
- Testing Items
  - Related Business Requirement
  - Steps to test
  - Expected result
  - Item pass or fail
  - Time and date tested

- Tester name
- Testing Notes
- Testing Schedule
  - Dependencies
  - Timelines
  - Schedule risks
- Process for fixing and retesting failed test items

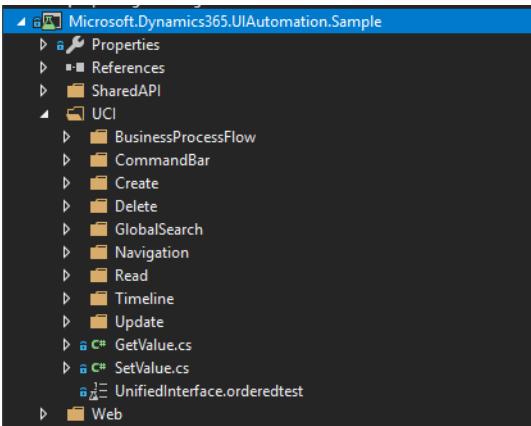
A good test plan describes the intention and scope of testing, guides the technical review process and supports a smooth rollout of functionality. A test plan should proceed user acceptance testing and have a means for tracking needed changes before rollout.

## User interface testing

EasyRepro is a library of tools from Microsoft to provide Dynamics customers the ability to facilitate automated UI testing for projects. The functionality provided covers the core commands that end users would perform on a typical workday and working to extend that coverage to more functionality. EasyRepro supports testing for Dynamics 365 versions 8.1 and higher.

## Sample Tests

Over 100 standard tests for the Web Client and over 50 tests for the Unified Interface are available. The sample test bank covers create, retrieve, update, and delete scenarios for multiple entities including contacts, opportunities, accounts, leads, and cases. This provides you the ability to easily run and start creating customized tests using the test bank as your initial code base. You can easily access the pre-defined tests within each of the specific functional areas you are testing.



## Coverage

### Functionality Covered

- Activity Row Wall
- Business Process Flow
- Charts

- Command bar
- Dashboards
- tables (Create, Update, Delete, Duplicate Detection, Assign)
- Global Search
- Guided Help
- Grids
- Sub-Grids
- Navigation
- Notifications
- Performance Center
- Quick Create
- Run Workflows
- Run Reports

## Functionality Not Covered

- Settings Area (Administration)
- Customization
- Advanced Find Query execution (Open Advanced Find is available)
- Export to Excel
- Templates
- Other Add-on Applications

Although there are not specific commands to cover the above functionality, you do have generic commands that will allow you to still interact with those features.

Though EasyRepro is available to all, using it requires the use of Visual Studio and will likely require developer assistance.

## Performance optimization

When analyzing performance of Microsoft Dataverse, it is important to understand areas affecting the performance of an app.

## Environment

By carefully examining various environmental factors you can improve your overall experience with Dataverse apps, even before your business begins to use it.

## Network

Network capacity can greatly affect performance of the apps from the end-user point of view. That includes web apps, Dynamics 365 for Outlook app, and custom applications using Web API.

There are two primary network characteristics affecting the performance: network bandwidth and latency. Collaboration with your network team is essential to making sure that network configuration is adequate and does not affect app performance.

## Client configuration

Dataverse apps are web-based applications and use web browsers as the user interface to view, add, or edit information that you've stored in the apps database. You may need to work with your system administrators to make sure that the client machines and software settings are optimized for performance.

## App Performance

Design of the app itself can greatly affect performance. One of the strengths of the Power platform is ability to quickly build robust small apps targeting specific users and specific functionality. Instead of a monolithic app that is more likely to suffer from performance issues you now have the ability to design several separate apps. This separation ensures that each app can be analyzed and tuned independently without too much impact on the others.

Design techniques for individual components may also affect performance of an app.

## Model-driven Apps

### Forms

To improve form performance, design them so that only the information essential for the users is included. If information required depends on the user's role, consider creating multiple forms instead of one monolithic form that includes every single column. Hiding tabs and sections that are not required to be immediately visible can also improve form responsiveness and performance.

One of the most common sources of performance issues on the forms is custom scripts added by developers. Good collaboration within the team is a key to identify and resolve any performance bottlenecks introduced during the development.

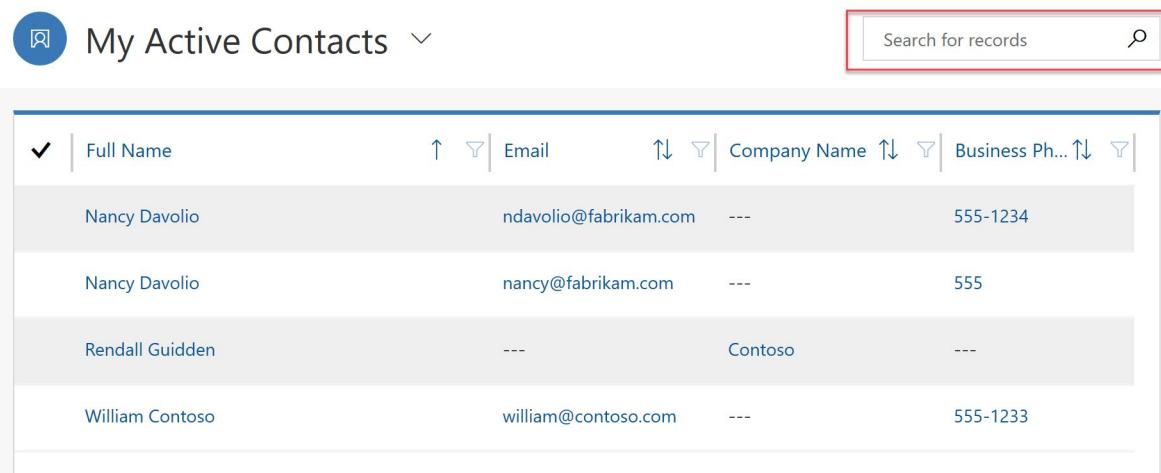
### Views

To improve performance of the table views, limit the number of columns displayed only to those required by the business. As often the case, it's a balance between the responsiveness and bringing enough information to improve usability of the view.

Customize your views so that they bring back an actionable number of rows. For example, Active Contacts view may contain hundreds of thousands of rows while My Active Contacts will limit that view only to the contacts that are owned by the user. At the same time consider limiting the number of rows per page so that the views return much more quickly.

### Quick search

Optimizing the quick search view only to include columns that makes sense to search will make a noticeable performance impact on searches within the views.



The screenshot shows a Microsoft Dataverse Quick View titled "My Active Contacts". At the top right is a search bar with the placeholder "Search for records" and a magnifying glass icon. Below the title is a table with four columns: "Full Name", "Email", "Company Name", and "Business Ph...". The table contains five rows of data:

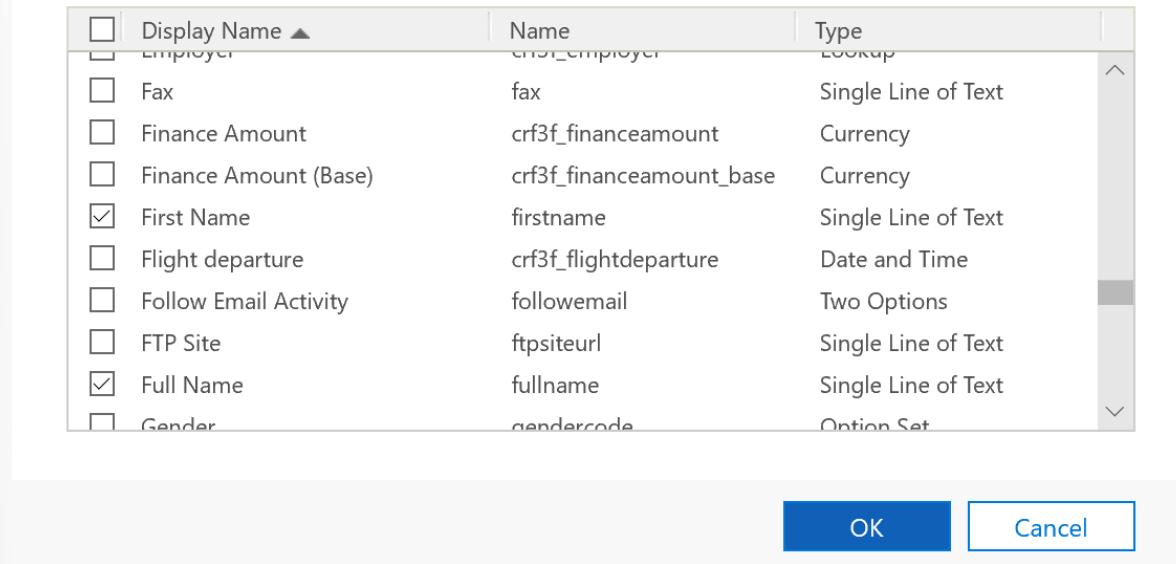
| Full Name       | Email                 | Company Name | Business Ph... |
|-----------------|-----------------------|--------------|----------------|
| Nancy Davolio   | ndavolio@fabrikam.com | ---          | 555-1234       |
| Nancy Davolio   | nancy@fabrikam.com    | ---          | 555            |
| Rendall Guidden | ---                   | Contoso      | ---            |
| William Contoso | william@contoso.com   | ---          | 555-1233       |

When a user searches for rows within a view, Dataverse performs a query that searches only predefined columns. Which columns are searched is defined by the Find columns settings in the system Quick View.

## Add Find Columns



Select the columns to be searched on for this Find view.



The screenshot shows the "Add Find Columns" dialog. It displays a list of columns from a table with three columns: "Display Name", "Name", and "Type". Some columns have checkboxes next to them, indicating they are selected for searching. The "Type" column includes a "Lookup" entry for the "Employee" column. The "Name" column lists internal column names like "crf3f\_employee", "fax", etc. The "Type" column lists data types like "Single Line of Text", "Currency", "Date and Time", etc. At the bottom are "OK" and "Cancel" buttons.

| Display Name                                   | Name                     | Type                |
|------------------------------------------------|--------------------------|---------------------|
| Employee                                       | crf3f_employee           | Lookup              |
| Fax                                            | fax                      | Single Line of Text |
| Finance Amount                                 | crf3f_financeamount      | Currency            |
| Finance Amount (Base)                          | crf3f_financeamount_base | Currency            |
| <input checked="" type="checkbox"/> First Name | firstname                | Single Line of Text |
| <input type="checkbox"/> Flight departure      | crf3f_flightdeparture    | Date and Time       |
| <input type="checkbox"/> Follow Email Activity | followemail              | Two Options         |
| <input type="checkbox"/> FTP Site              | ftpsiteurl               | Single Line of Text |
| <input checked="" type="checkbox"/> Full Name  | fullname                 | Single Line of Text |
| <input type="checkbox"/> Gender                | gendercode               | Option Set          |

OK      Cancel

The fewer number of columns that are selected, the faster quick search will be. Balance the number of find columns with the effectiveness of the search.

## Canvas Apps

The primary source of performance issues in canvas apps are poorly designed data access, excessive or suboptimal formula use, and overly complex user interface.

## Loading Data

Apps will perform poorly when most of the heavy lifting is done on the client and not on the server. Erroneous choice of how to query your data may lead to performance issues. Understanding **delegation<sup>29</sup>** in canvas apps is the key to improving you app performance when it comes to data access.

There are some canvas apps features that you would want to consider when optimizing your app start performance by using Concurrent function to load data from multiple sources concurrently.

## Controls Optimization

Another aspect that can affect performance is the number of screens and controls used in your app. Minimizing the number of controls and reducing complexity of the controls used can help boost your app performance. It will improve the performance while authoring the app as well. There are strategies to optimizing the number of controls used in your app. For example, you can use a gallery control instead of a Canvas/Data Cards when the data displayed is uniform or vary only slightly. Gallery can be powerful in reducing complexity, making your app easier to maintain.

There are other optimizations techniques available, make sure that canvas apps developers familiarize themselves with **detailed documentation<sup>30</sup>**.

## Workflows

Real-time workflows can be very effective but, if designed poorly, will affect the performance of the most common operations such as creating or updating a row. Consider converting poorly performing or long running workflows to background ones where the impact on the system will be smaller.

Background workflows create a log row when an instance of the workflow is executed. The default setting is to delete log entries for successfully executed workflows. If this setting is changed to keep the entries, jobs log will grow and may reach the point where it affects performance of the asynchronous services. Only use this flag for troubleshooting and avoid leaving it on in production.

## Plugins

Custom extensions such as plugins are the common source of the performance issues. Work with developers to identify and resolve those.

## Performance Testing

Designing for performance is an important part of the process and so is performance testing. It is important to perform performance testing on the system where the volume of data is close to your production system. That will give you a good understanding for app performance in general and for the areas to optimize. Use the instance copy function to create a copy of the production instance including the data.

The other important aspect of adequate performance testing is to reproduce workloads expected in the production environment. There is often a disconnect between testing environments where the system is accessed by a handful of the testers and the production instance where hundreds and thousands of users may access the system at the same time. There are load testing tools available from Microsoft and

<sup>29</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/delegation-overview>

<sup>30</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/performance-tips>

third-party vendors and you will need to collaborate with your development team to ensure that the tools are installed, configured, and used appropriately within the development lifecycle.

## Instance Statistics

Dataverse for Apps analytics are available via Power platform admin center <https://aka.ms/ppac>. It contains various statistics that can help identify potential areas for deeper performance analysis, for example most used tables, storage consumed, failing system jobs and plugins, etc.

## Query Performance

Dataverse is a service and, as any service, may not cater for your specific app design. Specific queries in your app may cause some performance issues. While you don't have direct access to the components of the Dataverse such as database, there are tools available within Dataverse to help in tuning the environment to your specific needs. You can analyze and optimize query performance using the Data Performance view discussed in detail in the next topic.

There are a number of tools that can assist in performance tuning various aspects of Power Apps.

## Dynamics 365 Diagnostics

There are two primary network characteristics affecting the performance: network bandwidth and latency. Identify if these may impact your app performance, by using the Dynamics 365 Diagnostics tool that is available at <https://<myorg>.crm.dynamics.com/tools/diagnostics/diag.aspx> where <myorg>.crm.dynamics.com is the URL of your Dynamics 365 organization.

**Dynamics 365 Diagnostics**

Diagnostic tests:

| Data Point                  | Action | Status   | Results Summary |
|-----------------------------|--------|----------|-----------------|
| Latency Test                |        | complete | 66 ms           |
| Bandwidth Test              |        | complete | 200 KB/sec      |
| Browser Info                |        | complete |                 |
| IP Address                  |        | complete |                 |
| JavaScript Array Benchmark  |        | complete | 21 ms           |
| JavaScript Morph Benchmark  |        | complete | 25 ms           |
| JavaScript Base64 Benchmark |        | complete | 5 ms            |
| JavaScript Dom Benchmark    |        | complete | 46 ms           |
| Organization Info           |        | complete | orga0fe228b     |
| All Tests                   | Run    | complete |                 |

Results:

```

Client Time: Mon, 21 Jan 2019 05:46:24 GMT
==== DOM Benchmark ====
Total Time: 46 ms
Breakdown:
Append: 10ms
Prepend: 12ms
Index: 2ms
Insert: 4ms
Remove: 18ms
Client Time: Mon, 21 Jan 2019 05:46:24 GMT
==== Organization Info ====
Organization name: orga0fe228b
Is Live: True
Server time: 1/21/2019 5:46:18 AM UTC
Url: https://practice102.crm6.dynamics.com/tools/diagnostics/diag.aspx
Client Time: Mon, 21 Jan 2019 05:46:24 GMT

```

**Clear** **E-Mail Results**

After opening the page and pressing Run, the report will become available that will assist your network team in detecting and resolving the issues. The statistic includes some browser benchmarks that may be useful in identifying if browser performance is an issue.

## Data Performance Logs

You can analyze and optimize query performance using the Data Performance view, which provides an aggregated list of tables with long-running queries. A long running query is defined as a query that takes three seconds or longer to complete. Typical examples of a component that can have a long running query is a plug-in with custom FetchXML or a sub-grid or view.

You can access Data Performance Logs via Settings - Administration - Data Performance.

The screenshot shows a table with the following data:

| Entity     | Count | Optimization Status    | Optimization Impact (%) |
|------------|-------|------------------------|-------------------------|
| systemuser | 480   | Optimization Available | 0.00                    |

At the bottom left, it says "1 - 1 of 1 (0 selected)". At the bottom right, there are navigation icons for "Page 1".

If one or more long running table queries are detected, log items are displayed in the view. You can use the Optimize command to apply optimizations to the selected query.

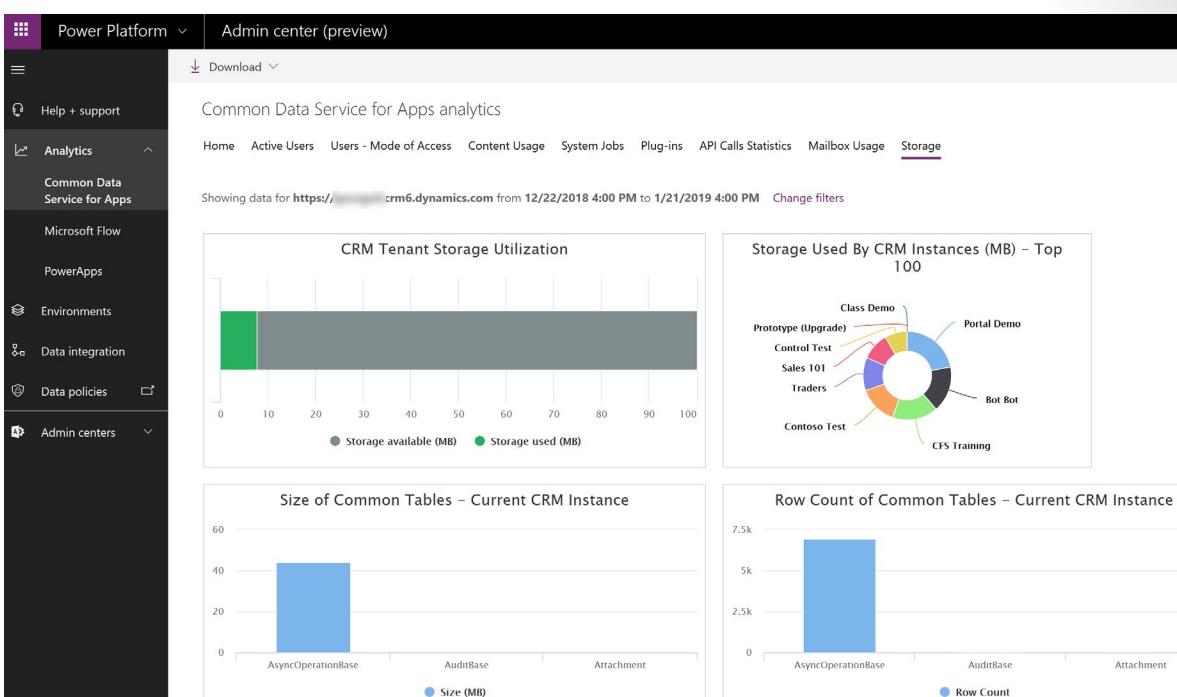
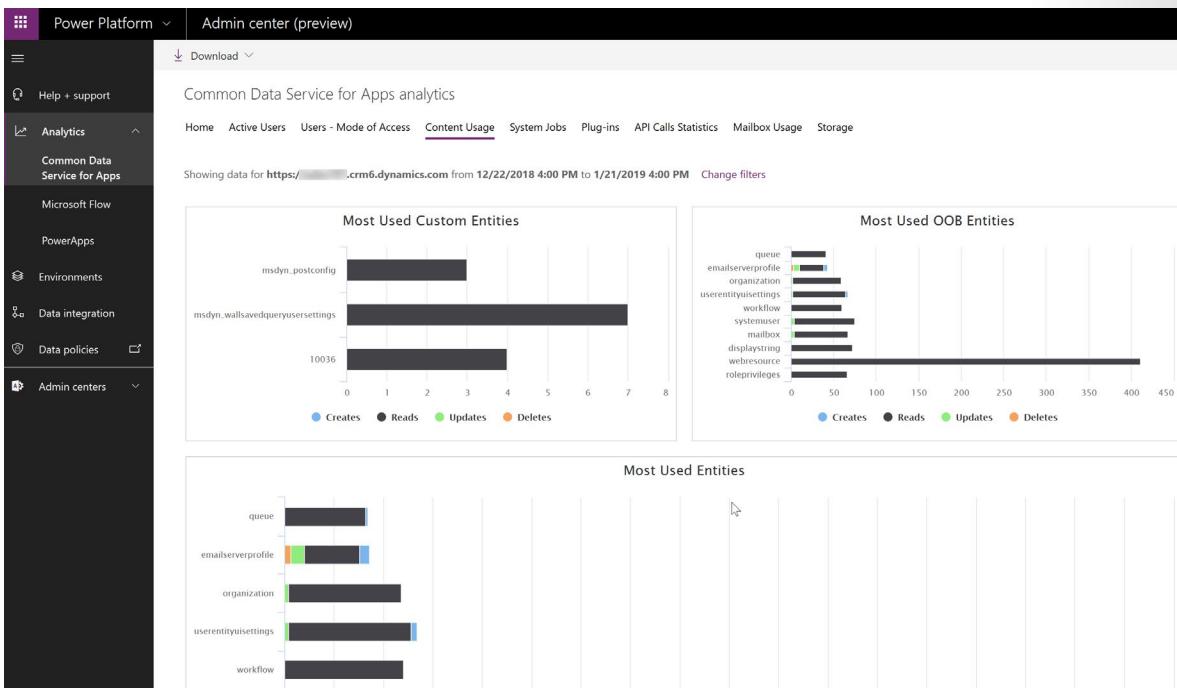
Note: With the most recent advances in Azure SQL database self-tuning capabilities you may no longer see any long running queries reported in the Data Performance view. Slow queries are automatically detected and optimized at the database level.

## Analytics

Dataverse analytics are available via the Power platform admin center <https://aka.ms/ppac>. Various statistics can help identifying potential areas for deeper performance analysis.

## Dataverse

The Content Usage tab contains information about most used out-of-the-box and custom tables. Identifying the most used tables helps concentrate the optimization efforts on the parts of the app that matter.



## Power Apps Analytics

Power Apps Analytics help with performance analysis by providing detailed statistics on Service Performance across various connectors used in your app.

Additional information such as overall app usage, location statistics, or errors seen by end users can be extracted from other tabs. While not directly performance related, this information can be useful in pinpointing overall problematic areas that may be caused by poor app performance such as specific region, specific app, etc.

## Documentation and the customer

Good documentation is key to the successful rollout of a Power Apps application. This video provides some best practices and suggestions on what to document for an app.



<https://www.microsoft.com/videoplayer/embed/RWs1Uz>

## summary

In this module, we covered testing plans, performance optimization and documentation.



<https://www.microsoft.com/videoplayer/embed/RWrMe8>

# Get started with Microsoft Dataverse for Teams

## Introduction

Microsoft Dataverse for Teams is a built-in, low-code data platform for Microsoft Teams that lets users build custom apps, chatbots, and flows in Teams by using Microsoft Power Apps, Microsoft Power Virtual Agents, and Microsoft Power Automate. Dataverse for Teams, which is built on Microsoft Dataverse, provides relational data storage, rich data types, enterprise-grade governance, and one-click solution deployment to the Microsoft Teams app. This module will introduce Dataverse for Teams and how you can build apps, workflows, chatbots, and dashboards in it. You will learn how to provision Dataverse for Teams and then create a table for your data.

The four types of Dataverse for Teams solutions are:

- **Apps** - Use Power Apps in Teams to create custom apps that are built by using the enhanced performance and scalability of Dataverse for Teams. Build custom solutions to meet your data and collaboration needs.
- **Workflows** - Automate repetitive tasks with Power Automate in Teams. Build flows that can respond to new Teams messages when data changes in Dataverse for Teams or when a Teams message is selected and has started a process. Bring more productivity to your Teams work.
- **Chatbots** - Respond to employee needs by building bots with Power Virtual Agents from within Teams. Share expertise through FAQs and interact with your data in Dataverse for Teams.
- **Dashboards** - Present powerful interactive charts and data with Power BI in Teams. Users can explore and interact with their data in Dataverse for Teams.

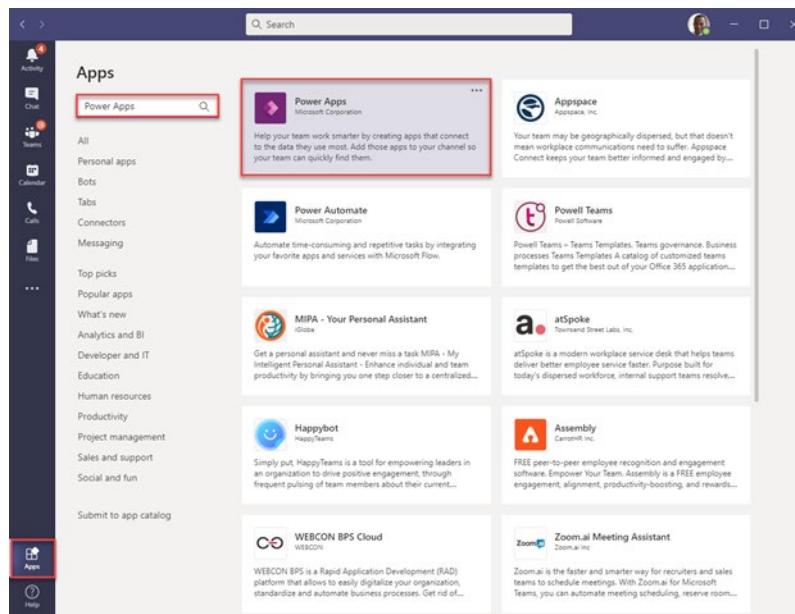
## Provision your first Dataverse environment

The Dataverse for Teams environment is where all business data, apps, and flows are stored. Each environment is associated to a Microsoft Office 365 group for the team. For this reason, each team can only have one Dataverse for Teams environment. When an app or chatbot is created or installed for the first time into a team, the Dataverse for Teams environment is provisioned automatically. After the environment has been created, you can create tables and flows in Dataverse.

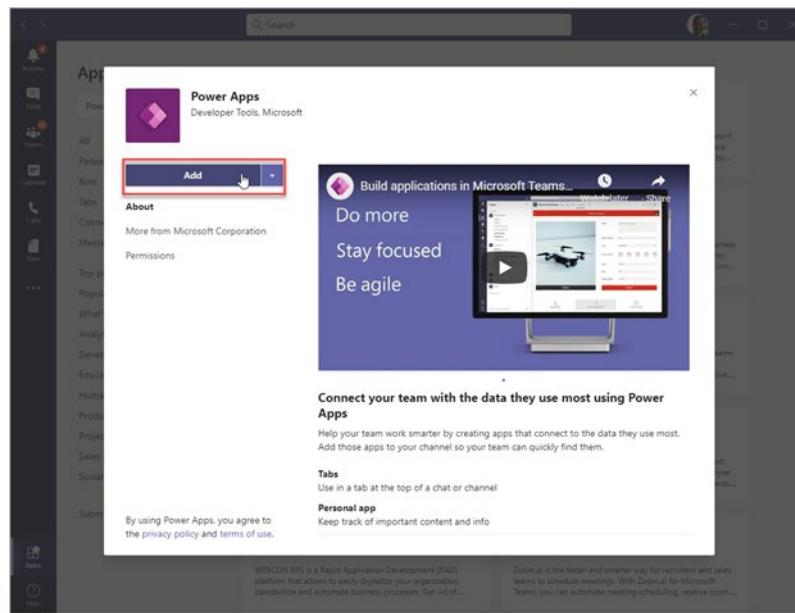
## Install Power Apps inside Teams

The Power Apps application for Teams lets you create, edit, and delete apps and tables in Dataverse for Teams. To provision the Dataverse for Teams environment, you first need to install an app into Teams.

1. To get started, open Teams and look for the **Apps** button in the lower-left corner of the window.
2. On the **App Marketplace** screen, use the search box to find Power Apps. Select the app labeled **Power Apps**.



3. Select the **Add** button to add the Power Apps application to your Teams client.

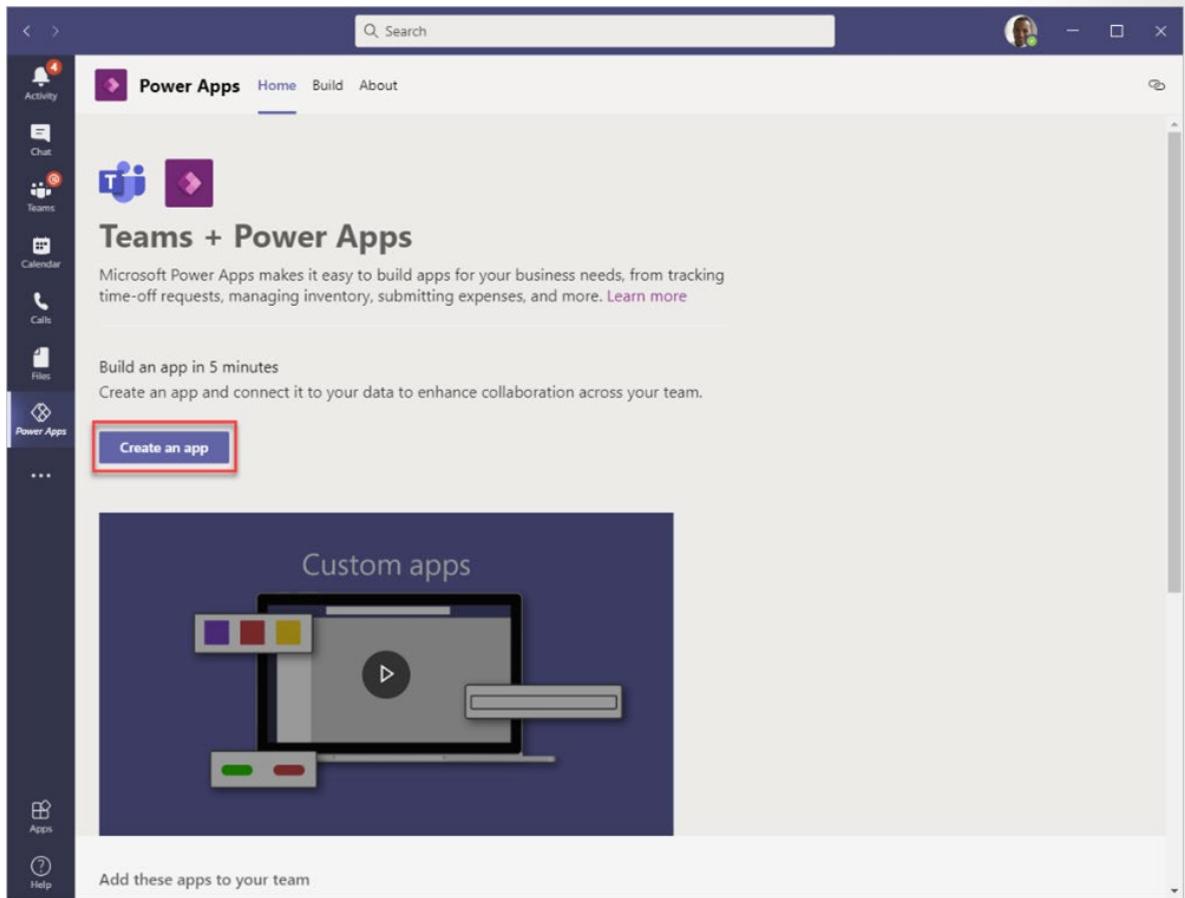


When Teams has successfully added Power Apps, you will be brought to the main **Power Apps** screen. This screen provides access to existing apps along with templates, videos, and other resources on building apps.

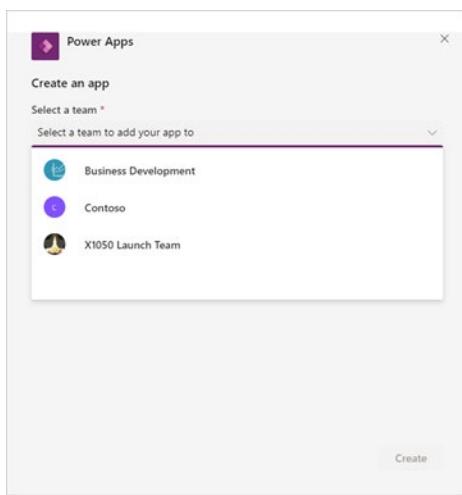
## Provision Dataverse for Teams to create a table

Now that you have the Power Apps application installed, you can create the first app, which will automatically provision the environment. Decide on which team to create the environment and to store your table. The Summary unit of this module provides links if you need to create a new team.

1. Select the **Create an app** button.

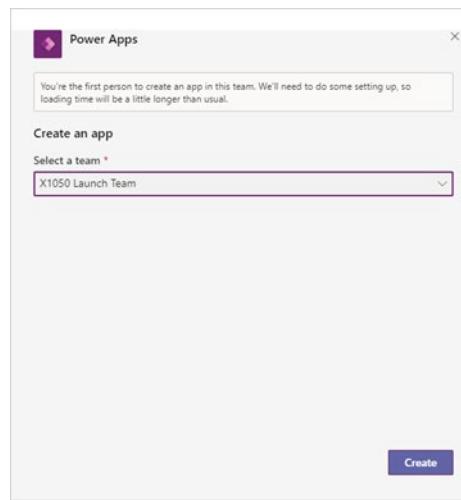


2. Select the team for your app.

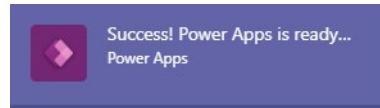


A message will appear, stating that you are the first person to create an app in this team.

3. Select **Create** to have the Dataverse for Teams environment built. While waiting for the process to complete, you can close the window and do other work.

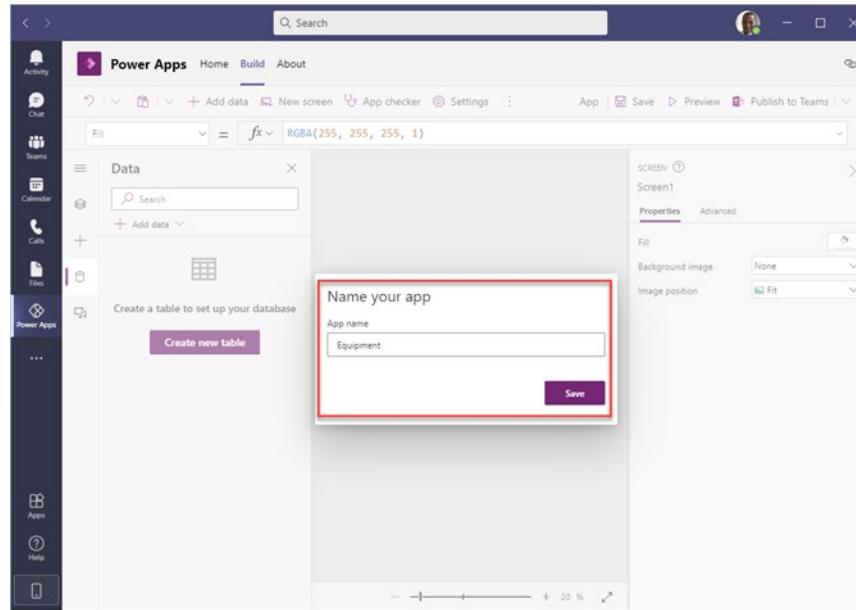


When your environment is provisioned, a pop-up window will display in the lower-right corner of your screen, letting you know that the process has finished.



At this stage, the Power Apps editor will appear.

4. Enter the name for your app and then select **Save**.



With the creation of the first app, you now have a Dataverse for Teams environment to help you start building tables. The Power Apps editor enables you to quickly create tables with the **Create new table** button, which the next section will explain.

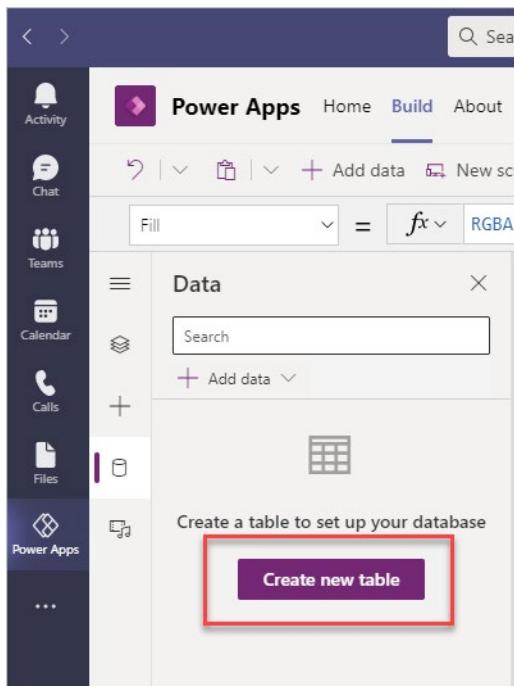
# Create your first table to store data

Now that you have provisioned a Dataverse for Teams environment for the team that you specified, you can begin creating solutions. Power Apps, Power Automate flows, and chatbots are ineffective without data to drive them, so the first step in this process is to create a table within your environment to store your data.

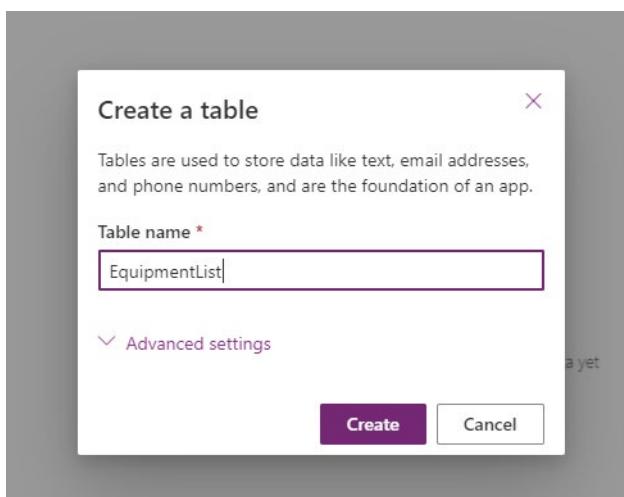
## Power Apps visual table editor

To access the Power Apps visual table editor, follow these steps:

1. While editing an app in the Power Apps editor for Teams, select the **Create new table** button.



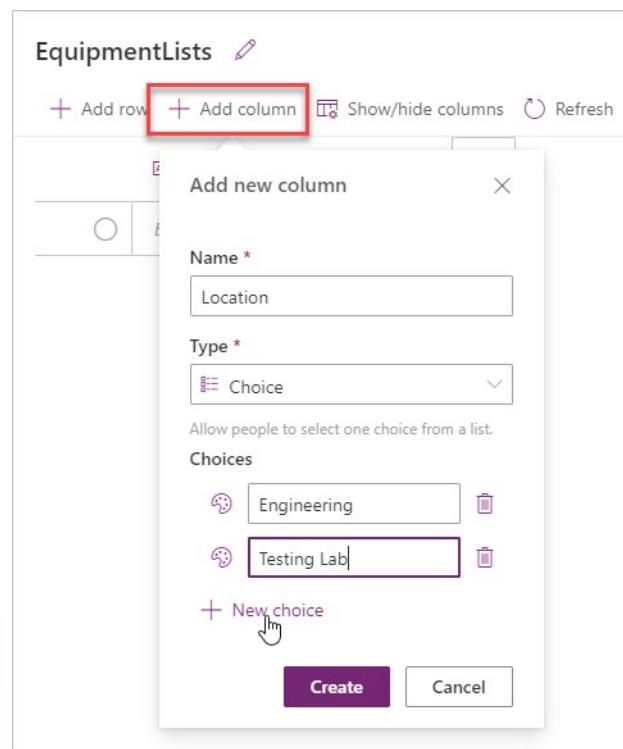
2. Enter **EquipmentList** in the **Table name** column because you want the table to hold a list of equipment. Select **Create** to complete the process.



After a few seconds, your table will be provisioned and will display in the visual table editor. The visual table editor enables you to quickly populate your table with the **+ Add column** and **+ Add row** options.

To exemplify the process of using the visual table editor to populate your table, the following steps show how to add a **Location** column that will have two choices of **Engineering Bay** and **Testing Lab**.

3. Select **+ Add column**.
4. In the **Name** column, enter **Location**.
5. To indicate the type of data that the table will contain, select **Choice** from the **Type** drop-down menu.
6. Fill in the **Choices** options with **Engineering Bay** and **Testing Lab**. To add more choices, select **+ New choice**.
7. Select **Create** when finished.



From the visual table editor, you can use any of the following data types. The Summary section at the end of this module provides links for a complete list of available data types and how they work.

| Data type   | Description                                                 |
|-------------|-------------------------------------------------------------|
| Text        | A single line of text                                       |
| Email       | Text in email format; will be selectable for the user       |
| URL         | A hyperlink in URL format; will be selectable for the user. |
| Auto Number | Defines an autogenerated number sequence                    |
| Number      | An integer (for example, 10 or -10)                         |
| Date        | Allows users to enter or search for a date                  |

| Data type | Description                                              |
|-----------|----------------------------------------------------------|
| Decimal   | A number with a decimal point (for example, 1.5 or -1.5) |
| Lookup    | References a primary column in another table             |
| Choice    | Allows people to select one choice from a list           |
| Yes/No    | Allows people to pick between two choices                |

8. Select **+ Add rows** to fill out your table with data. The input for the columns depends on the type of column that you selected. For this example, two choices are available in the **Location** drop-down list.

The screenshot shows the 'EquipmentLists' table in the Microsoft Dataverse for Teams visual table editor. The 'Name' column has a row with 'Digital Whiteboard'. The 'Location' column has a dropdown menu open, showing three options: 'Select Option', 'None', 'Engineering Bay', and 'Testing Lab'. The 'None' option is currently selected. The editor interface includes standard table management buttons like 'Insert row above', 'Insert row below', and 'Delete 1 record(s)'.

9. After you have finished editing the table, select **Close** from the bottom of the editor to return to the app. Now that you have data for the app, Dataverse for Teams automatically generates a basic application by using the Hero Template.

## Power Apps table editor

Occasionally, you will want to edit your table and include more data types that are not in the visual table editor.

1. To open your table in the full editor, select **Build** and then select the **See all** option from the Power Apps application.

The screenshot shows the Microsoft Power Apps Build interface. At the top, there's a navigation bar with 'Power Apps', 'Home', 'Build' (which is highlighted with a red box), and 'About'. Below that is a team header for 'X1050 Launch Team' with a team icon and the team name. Underneath, there are two tabs: 'Built by this team' (which is selected) and 'Installed apps'. A search bar is at the top right. The main area is titled 'Items created for X1050 Launch Team' and lists two items: 'Equipment' and 'EquipmentList'. At the bottom of this list is a 'See all' link, which is also highlighted with a red box.

2. Select **Tables** and then select the ellipsis (...) for the table. Selecting **Edit** will bring up the full table editor, or you can select **Edit data** for the visual table editor experience. For now, select **Edit** for the full editor.

This screenshot shows the 'Tables' section of the Microsoft Power Apps Build interface. On the left, there's a sidebar with icons for Activity, Chat, Teams, Calendar, Calls, Files, and Power Apps. The 'Tables' option is selected and highlighted with a red box. The main area shows a list of tables under 'X1050 Launch Team > Tables'. One table, 'EquipmentList', is selected and highlighted with a red box. An ellipsis menu is open next to it, with a red arrow pointing to the 'Edit' option, which is also highlighted with a red box. Other options in the menu include 'Delete table', 'Data', and 'Edit data'.

At first, it might appear as if a mistake has been made because several columns that you did not create are displayed. This result is normal and an example of the power of Dataverse in action.

The screenshot shows the Microsoft Power Apps interface, specifically the 'Tables' section. The left sidebar has a dark theme with icons for Activity, Chat, Teams, Calendar, Calls, Files, and Power Apps. The main area shows the 'EquipmentList' table from the 'X1050 Launch Team' project. The table has 21 columns:

| Display name               | Name              | Data type  | Type     | Custom... | Required | Searchable... |
|----------------------------|-------------------|------------|----------|-----------|----------|---------------|
| Created By                 | createdby         | Lookup     | Standard | ✓         | Optional | ✓             |
| Created By (Delegate)      | createdonbeh...   | Lookup     | Standard | ✓         | Optional | ✓             |
| Created On                 | createdon         | Date an... | Standard | ✓         | Optional | ✓             |
| EquipmentList              | crfae_eqipment... | Unique ... | Standard | ✓         | Required | ✓             |
| Import Sequence Number     | importsequenc...  | Whole ...  | Standard | ✓         | Optional | ✓             |
| Location                   | crfae_location    | Choice     | Custom   | ✓         | Optional | ✓             |
| Modified By                | modifiedby        | Lookup     | Standard | ✓         | Optional | ✓             |
| Modified By (Delegate)     | modifiedonbeh...  | Lookup     | Standard | ✓         | Optional | ✓             |
| Modified On                | modifiedon        | Date an... | Standard | ✓         | Optional | ✓             |
| N... (Primary Name Column) | crfae_name        | Text       | Custom   | ✓         | Required | ✓             |
| Owner                      | ownerid           | Owner      | Standard | ✓         | Required | ✓             |
| Owning Business Unit       | owningbusines...  | Lookup     | Standard | ✓         | Optional |               |
| Owning Team                | owningteam        | Lookup     | Standard | ✓         | Optional |               |
| Owning User                | owninguser        | Lookup     | Standard | ✓         | Optional |               |
| Record Created On          | overridendate...  | Date Only  | Standard | ✓         | Optional | ✓             |
| Status                     | statecode         | Choice     | Standard | ✓         | Required | ✓             |
| Status Reason              | statuscode        | Choice     | Standard | ✓         | Optional | ✓             |
| Time Zone Rule Version ... | timezonerulev...  | Whole ...  | Standard |           | Optional |               |
| UTC Conversion Time Zo...  | utcconversion...  | Whole ...  | Standard |           | Optional |               |
| Version Number             | versionnumber     | Big Int... | Standard |           | Optional |               |

## Different available column types

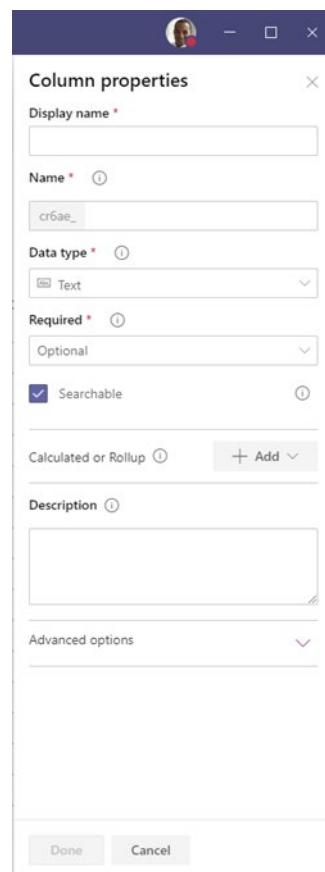
The other columns in your table are created by the system. Those columns are useful if you need more information about your data. For example, the **Created On** column can tell you when a record was created, while the **Modified By** column can tell you who last touched it.

Likely, you will want to add other columns. The following steps will show you the options that you have.

1. Select the **+ Add column** button at the top of the screen.

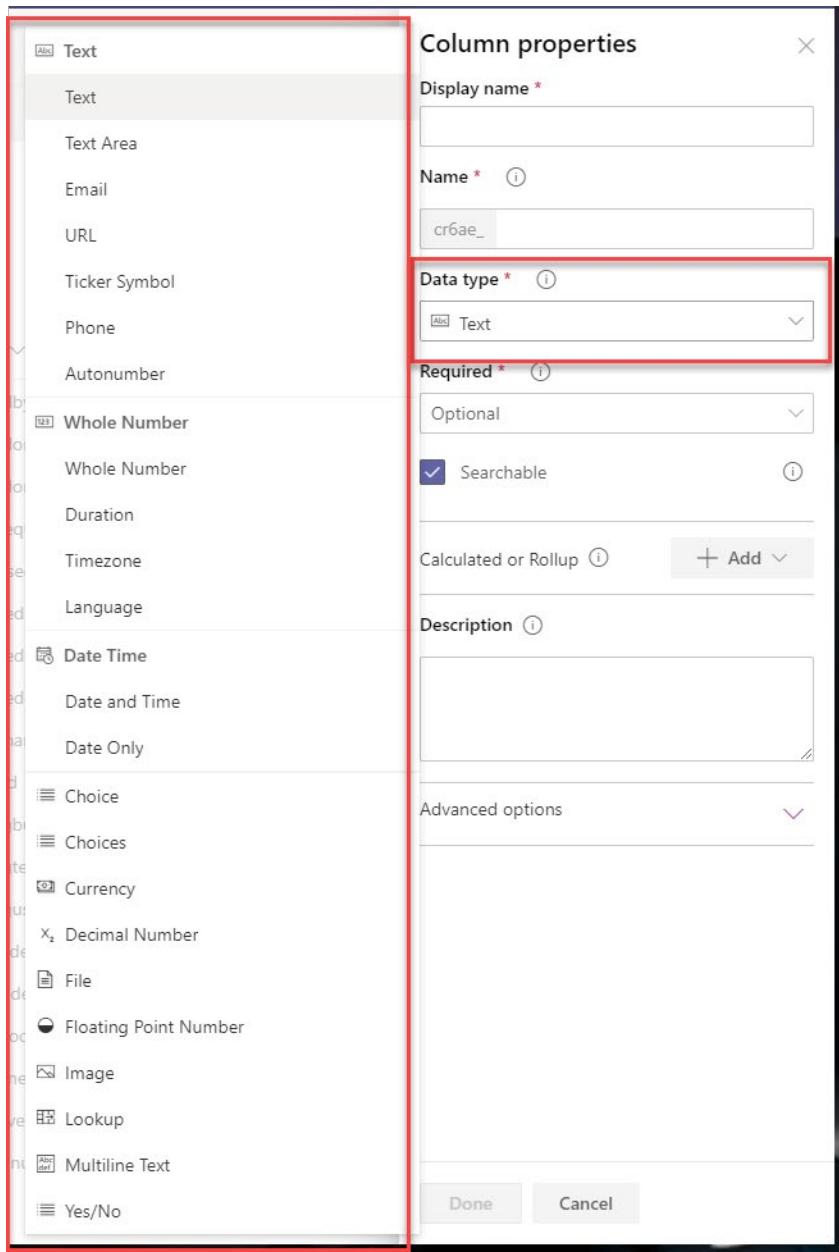
The screenshot shows the Microsoft Power Apps interface, specifically the 'Tables' section. The left sidebar has a dark theme with icons for Activity, Chat, Teams, Calendar, Calls, Files, and Power Apps. The main area shows the 'EquipmentList' table from the 'X1050 Launch Team' project. The '+ Add column' button in the top navigation bar is highlighted with a red box. The table structure is identical to the one in the previous screenshot.

A blade will appear on the right side of the screen. The **Display name** of your column is what your users would see if they were entering data directly into the table, while the **Name** column shows what the column is named in the database.



The **Data type** column is perhaps the most important part of designing a column because it defines what type of data is being held within the column.

2. Select the **Data type** drop-down menu to show the full range of available data types. Several options are available for you to choose from; the most common ones are explained in the following table. Each data type has special characteristics and settings that can be applied on selection.



| Data type | Description                                                                                                                                                                |
|-----------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Text      | Text is similar to a <b>Single Line of Text</b> column that you might see in a SharePoint list, though this data type in Dataverse can hold large strings of text as well. |

| Data type     | Description                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Email         | Email would be used if you knew that this column should only hold email addresses. By using a data type that is aligned with the data that you'll be entering into the column, you will have access to intelligent, out-of-the-box features that are specific to the type of data that you're working with. Email records, for example, are selectable and will launch Outlook or other mail clients to allow the user a more streamlined way to send an email to someone. |
| Date and Time | Date and Time should be used when the time of day is important to a record, while Date Only can be used if only the date is relevant.                                                                                                                                                                                                                                                                                                                                      |
| Whole Number  | Whole number is used for numbers that don't have decimal points, while Decimal Number is used for numbers that do.                                                                                                                                                                                                                                                                                                                                                         |
| Currency      | If you are keeping track of dollar figures in your table, a Currency data type might be relevant for one of your columns.                                                                                                                                                                                                                                                                                                                                                  |
| Image         | The Image data type exists specifically to hold images, which might be valuable if you plan to use images in the Power Apps application that uses this table as a data source.                                                                                                                                                                                                                                                                                             |

The Summary unit of this module provides links that you can select for a complete list of these data types, their characteristics, limitations, and more.

## Management interface overview

Now that you have provisioned your environment and have created a table with columns of your choosing, you can set administrative settings on your table. Dataverse for Teams includes a management interface for you to complete that task, if you are a team owner. Members of a team can't change the permissions.

1. While editing your table within the Power Apps **Build** screen, select **Manage Permissions** from the ribbon menu, which is highlighted by a red box in the following screenshot. This option does not appear if you are not a team owner.

The screenshot shows the 'Manage permissions' screen for the 'EquipmentList' table in the 'X1050 Launch Team > Tables' section. The table contains 17 columns with various data types and properties. The columns are:

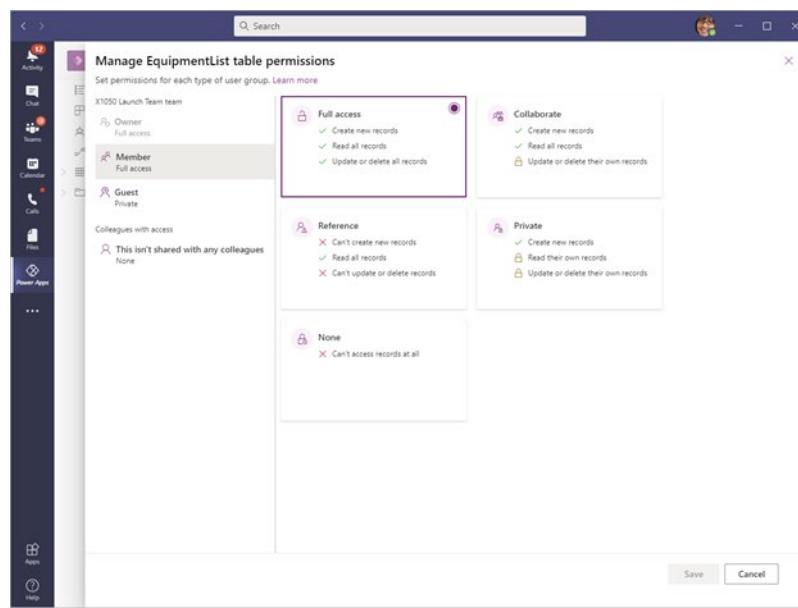
| Display name              | Name             | Data type  | Type     | Custom... | Required | Searchable... |
|---------------------------|------------------|------------|----------|-----------|----------|---------------|
| Created By                | createdby        | Lookup     | Standard | ✓         | Optional | ✓             |
| Created By (Delegate)     | createdonbeh...  | Lookup     | Standard | ✓         | Optional | ✓             |
| Created On                | createdon        | Date an... | Standard | ✓         | Optional | ✓             |
| EquipmentList             | crbae_equipm...  | Unique ... | Standard | ✓         | Required | ✓             |
| Import Sequence Num...    | importsequenc... | Whole ...  | Standard | ✓         | Optional | ✓             |
| Modified By               | modifiedby       | Lookup     | Standard | ✓         | Optional | ✓             |
| Modified By (Delegate)    | modifiedonbe...  | Lookup     | Standard | ✓         | Optional | ✓             |
| Modified On               | modifiedon       | Date an... | Standard | ✓         | Optional | ✓             |
| N (Primary Name Column)   | crsae_name       | Text       | Custom   | ✓         | Required | ✓             |
| Owner                     | ownerid          | Owner      | Standard | ✓         | Required | ✓             |
| Owning Business Unit      | owningbusines... | Lookup     | Standard | ✓         | Optional |               |
| Owning Team               | owningteam       | Lookup     | Standard | ✓         | Optional |               |
| Owning User               | owninguiser      | Lookup     | Standard | ✓         | Optional |               |
| Record Created On         | overridendate... | Date Only  | Standard | ✓         | Optional | ✓             |
| Status                    | statecode        | Choice     | Standard | ✓         | Required | ✓             |
| Status Reason             | statuscode       | Choice     | Standard | ✓         | Optional | ✓             |
| Time Zone Rule Version... | timezonerulev... | Whole ...  | Standard |           | Optional |               |
| UTC Conversion Time ...   | utcconversion... | Whole ...  | Standard |           | Optional |               |

- On the **Manage permissions** screen, you can define who has access to your table and its data. Different privilege levels are available for the team and colleagues who have this table shared with them.

Five different permission options include:

- Full access** - Allows users and groups full access to create new records, read or update existing records, and delete records. This level of access should only be granted to users who need full control over the data.
- Collaborate** - Allows the creation of new records and the ability to read existing records, but it denies the ability to update existing records or delete them unless they were created by the user. This level of privilege is beneficial for typical users who shouldn't have administrative level access over the data that they are connecting with, but who still need the ability to create new records in the table and edit or delete records that they created in the past. This level of privilege is the most commonly assigned for most use cases.
- Reference** - Allows the assigned user or group the ability to read from the table but not to interact with the data in any other way. For example, this level of privilege is useful for look-up tables where an administrator should define the data that exists while the users should only be able to read from it as a reference for their work.
- Private** - A unique privilege level because it allows the users or groups to create new records in the table (which **Reference** does not), but it does not allow users or groups to read records that were not created by them. It also allows the user or group to update and delete their own records, but not records that were created by other users. This level of privilege is best used in situations where sensitive information is held within the table that should not be exposed to typical users.
- None** - Denies access entirely; the user or group can't view or modify records within the table and they can't delete them.

All levels of privilege can be assigned to either users or groups, depending on your requirements. It is not uncommon to add guest users to a team, and those users might need to have a different level of access than internal organization users. For this reason, the **Manage table permissions** section will allow you to manage members separately from guests by using the options on the left blade.



3. Select the user or group that you want to assign access to, and then select the level of access that they should have. If your team has multiple owners, the **Owners** option will be visible, and you can define access for those users as well. In the preceding screenshot, members of the team have been granted **Full access**.

## Summary

This module explained how Microsoft Dataverse for Teams is defined as a product, how to provision a Dataverse for Teams environment, how to create your first table within the environment, and how to manage permissions on the table. Though this module has provided only an overview of Dataverse for Teams, it's an excellent start to helping you discover the possibilities that this new low-code, no-code data platform provides when you want to create customized solutions to your business problems. The next modules closely examine the platform and explore building a Power Apps application within the new environment that you created. Additionally, the next modules explain how to connect chatbots, Microsoft Power BI reports, and Power Automate flows to create a robust solution that is capable of changing the way that your users interact with their data.

## Links to related modules for specific topics

| Topic                                                           | Link          |
|-----------------------------------------------------------------|---------------|
| Additional information on the available data types in Dataverse | Data types    |
| Information on how to create teams and channels                 | Create a team |

# Build your first app with Power Apps for Teams

## Introduction

Today's business problems increasingly require modern digital solutions. With a low-code platform, anyone with a great idea can build a digital app. Microsoft Dataverse for Teams allows you to build modern digital apps and deploy them to help you and your team solve those problems.

Each app that you build uses the hero template as the default app layout to help you quickly get your app working. It allows you to list records, create new records, and then drill into existing records to review or make changes.

| Equipment                     |                 |                   |
|-------------------------------|-----------------|-------------------|
| <a href="#">New record</a>    | Name            | Service Needed?   |
| Video Projector<br>1/1/2020   | Video Projector | Yes               |
| Conference phone<br>1/1/2020  | Notes           | Needs new bulb    |
| Whiteboard Tripod<br>1/1/2020 | Date Deployed   | Location          |
|                               | 1/1/2020        | Conference Room B |

This module explains how to use Dataverse for Teams to build an app.

## Create your first app with the hero template

The following steps show how to create an app from Microsoft Teams.

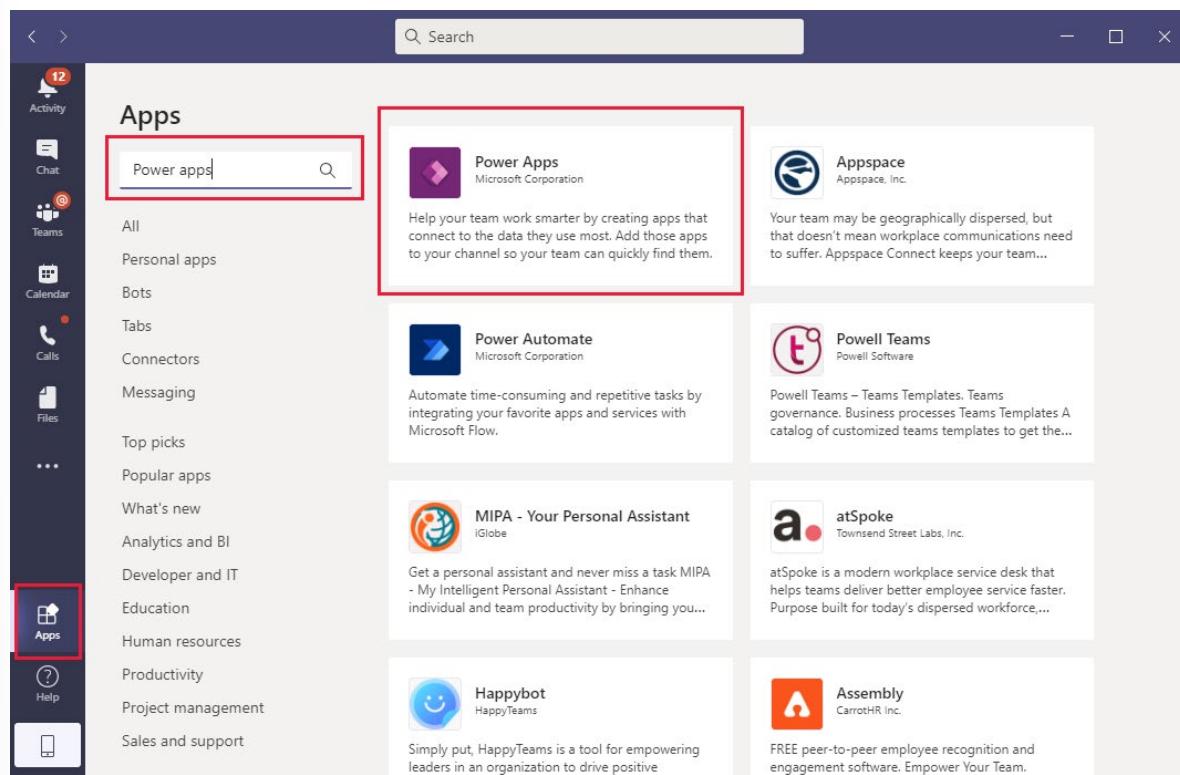
### Step 1: Identify a Microsoft team

All Dataverse for Teams apps require a Microsoft team to store the Microsoft Power Apps data. Only team members will be able to access your app by default.

### Step 2: Add the Power Apps application to Teams

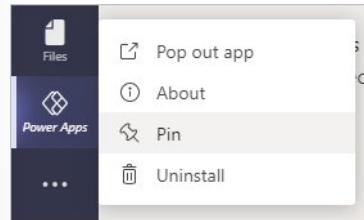
1. Launch Teams or use your browser to open [Teams<sup>31</sup>](#).
2. In Teams, select **Apps** in the lower-left corner of the screen.
3. In the search box, enter **Power Apps** and then select it from the search results.

<sup>31</sup> <https://teams.microsoft.com>



1. Select **Open** to install Power Apps.

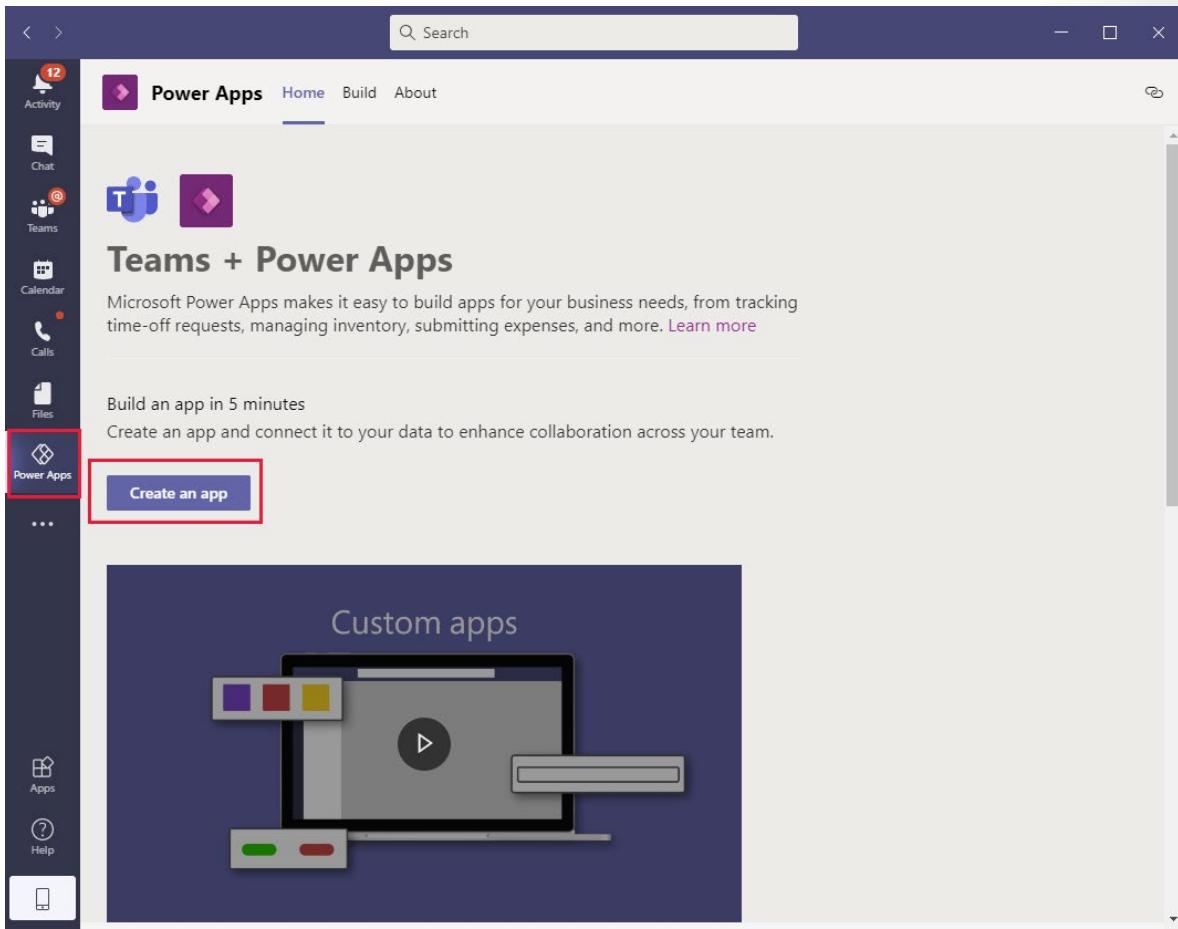
Note: Pin the app for Power Apps to the app launcher so that you can quickly access it any time. Right-click the Power Apps icon and then select \*\*Pin\*\*. You can also run the app as a separate window from Teams by selecting \*\*Pop out app\*\*.



## Step 3: Create a new app

After you have installed Power Apps, you can create your first app.

1. From Teams, open the Power Apps application and then select **Create an app**.



1. Select a team for your app and then select **Create**.
2. Enter the name of your app and then select **Save**.

The hero template for your app will display. It's currently empty because you haven't connected a Dataverse table to it.

## Step 4: Create and connect a table to your app

1. Open your new app and then select **Create new table**.
2. Enter a new table name and then select **Create**.
3. Add new columns to your table by selecting **+ Add column**.

For each column, provide a **Name** and then select the **Type** of data that it holds. By choosing the appropriate data type, you will be able to access intelligent, out-of-the-box features to make your app more manageable for people to use.

For the table, you need to track the date that the equipment was deployed, the location, whether service is needed, and relevant notes.

- For tracking the deployment date, use date type of **Date**.

Add new column X

Name \*

Type \*  
 Date

Allow users to enter or search for a date.

Create Cancel

- To limit the location options to "Conference Room A" and "Conference Room B," use the **Choice** data type. To add more choices, select **+ New choice**.

Add new column X

Name \*

Type \*  
 Choice

Allow people to select one choice from a list.

Choices

Conference Room A Conference Room B

+ New choice

Create Cancel

- To indicate if service is needed, use **Yes/No** as the data type because you only need to switch between two options. This data type limits a user to only two choices.

The screenshot shows the 'Add new column' dialog box. The 'Name \*' field contains 'Service Required?'. The 'Type \*' dropdown is set to 'Yes/No'. Below it, a note says 'Allow people to pick between two choices.' Under 'Items', there are two radio buttons: 'Yes' (selected) and 'No'. The 'Default value' dropdown is set to 'No'. At the bottom are 'Create' and 'Cancel' buttons.

- Use the **Text** data type for the **Notes** column.

The screenshot shows the 'Add new column' dialog box. The 'Name \*' field contains 'Notes'. The 'Type \*' dropdown is set to 'Text'. Below it, a note says 'A single line of text.' Under 'Advanced options', there is a dropdown menu. At the bottom are 'Create' and 'Cancel' buttons.

The following list of data types are available by using the visual table editor. For a complete list of all available data types and how they work, see the links in the Summary unit at the end of this module.

| Data Type   | Description                                                 |
|-------------|-------------------------------------------------------------|
| Text        | A single line of text.                                      |
| Email       | Text in email format; will be selectable for the user.      |
| URL         | A hyperlink in URL format; will be selectable for the user. |
| Auto Number | Defines an autogenerated number sequence.                   |
| Number      | An integer (for example, 10 or -10).                        |
| Date        | Allows users to enter or search for a date.                 |
| Decimal     | A number with a decimal point (for example, 1.5 or -1.5).   |
| Lookup      | References a primary column in another table.               |

|  |        |                                                 |
|--|--------|-------------------------------------------------|
|  | Choice | Allows people to select one choice from a list. |
|  | Yes/No | Allows people to pick between two choices.      |

4. To add data to your table, select **+ Add row**. Enter data that you want people to see when they use your app.

Notice how the choice of data type changes the user experience to fit the type of data that is being stored.

- **Date Deployed** - A calendar can be used to select the **Date**.
- **Location** - Uses a drop-down list of options that are defined in the **Choice** data type.
- **Service Required** - The user can select between **Yes/No** with a toggle.
- **Notes** - The user can enter **Text**.

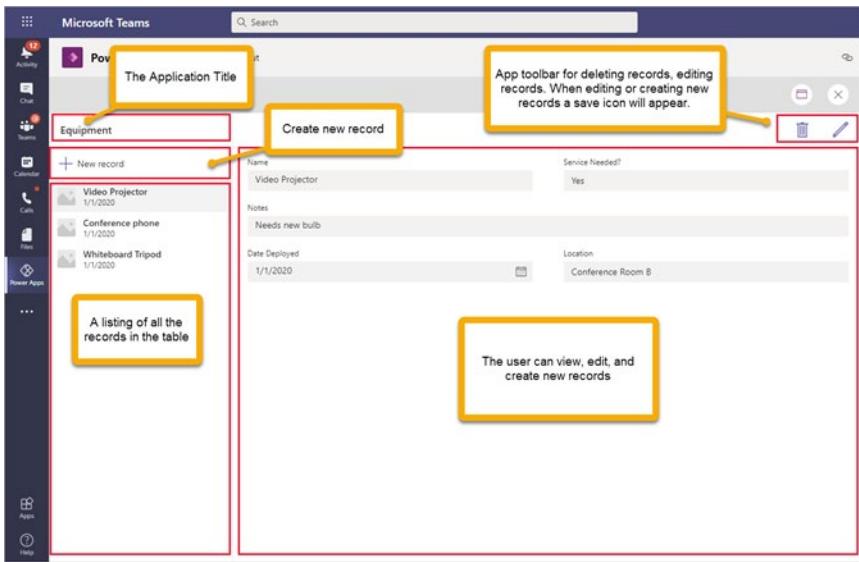
The screenshot shows a Microsoft Power Apps table editor titled "Equipment". The table has five columns: "Name", "Date Deployed", "Location", "Service ...", and "Notes". The "Date Deployed" column contains a date picker showing "March 2021" with the 16th highlighted. The "Service ..." column has a "No" toggle button. The "Notes" column has a text input field with "Enter text" placeholder. The toolbar at the top includes buttons for "Insert row above", "Insert row below", "Delete 1 record(s)", and a plus sign for adding a new row.

1. Select **Close** to finish editing the table. It might take a few minutes to process the table. After the processing has completed, the hero template will appear, showing the data loaded into the app.
2. Select **Save** in the toolbar to create the first saved version. After the first save, Power Apps will continue to save automatically every two minutes while you are editing.

## Step 5: Preview and test your app

To preview and test your app, select the **Preview** button in the upper right.

The hero template automatically creates a functioning app with your data. After a data source has been linked, you can instantly view, edit, and delete existing records or create new ones.

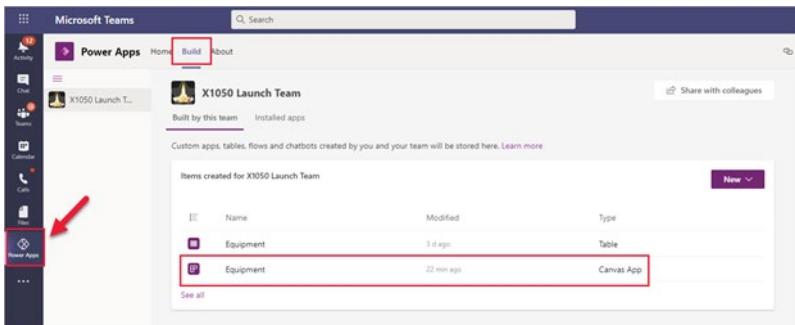


## Customize your app with Power Apps Studio

In the previous lesson, you created a new app by using your data that was loaded into Dataverse for Teams. The hero template can quickly make a useful app, but you will often want to customize the app to suit your needs.

For this lesson, you will change the title of the app, the layout of the gallery, and the order of fields.

To start, open Teams by using the desktop client or by browsing [Teams<sup>32</sup>](#), and then open the app for Power Apps. Select the **Build** tab at the top and then select your app.



Your app will open in Microsoft Power Apps Studio for editing.

### [!NOTE]

It's always a good idea when you first start editing to select **Save** in the toolbar. After the first save, Power Apps will continue autosaving every two minutes while you are editing.

Power Apps applications are built by using various UI elements that are referred to as **Controls**. Use controls to help create a better experience when your users are navigating and interacting with the app.

Common controls include:

- **Label** - Display information to the user as text, numbers, dates, or currency.
- **Edit form** - Allows the user to create and edit records and then save them.

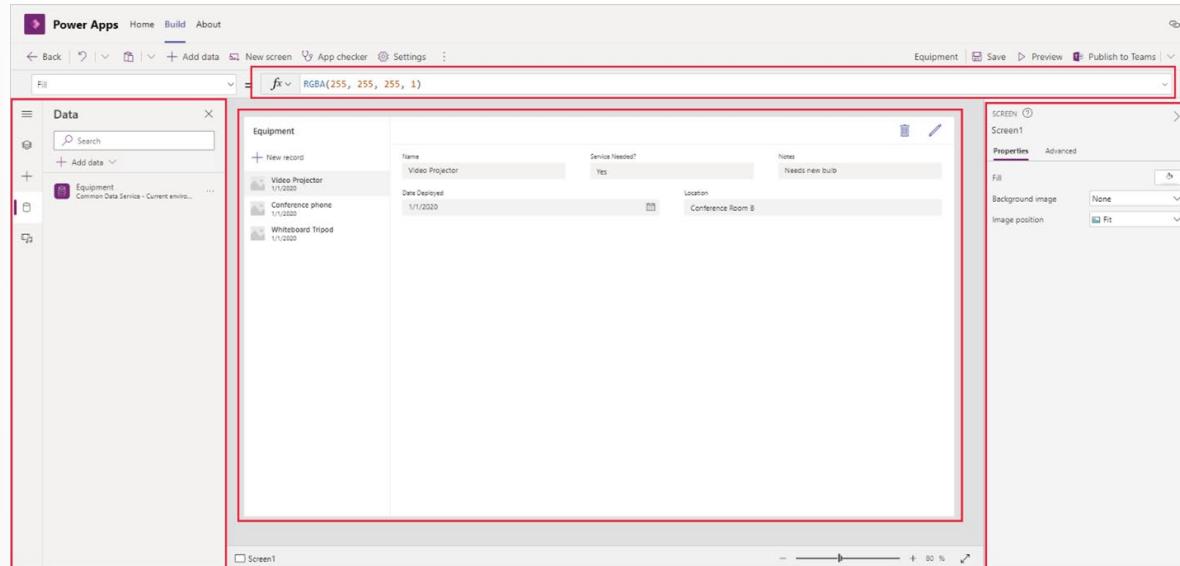
<sup>32</sup> <https://teams.microsoft.com/?azure-portal=true>

- **Text box** - A box where a user can enter data such as text and numbers.
- **Vertical gallery** - Displays multiple records from a data source in a vertical orientation. This control can show multiple types of data for each record.
- **Add icon** - Select from a library of graphical symbols. This control can be configured to respond when a user selects them.
- **Rectangle** - A border shape that can be placed anywhere on the app.
- **Date picker** - Allows the user to select a date by using a calendar pop-up window.
- **Button** - A way for the user to interact with the app.

When you are editing the app, select between the following options in the left column:

- **Tree view** - Displays a visual hierarchy of all controls in the app.
- **Insert** - Insert new controls such as labels, buttons, icons, and forms.
- **Data** - Add more tables from Dataverse for Teams or add a connector to other Microsoft Office 365 services.
- **Media** - Add images, videos, and audio.

The center of the screen is where you move and edit the components that make up your app. When a control is selected from the tree view or in the app view, the **Properties** pane displays the properties. At the top of the screen is the formula bar, which allows for more editing capabilities.



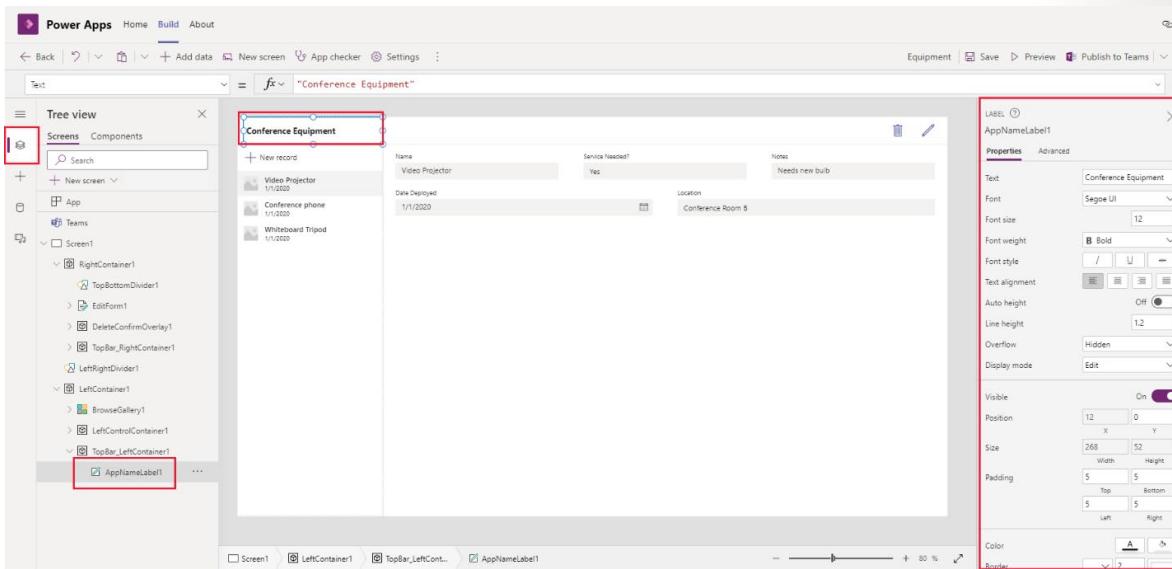
## Change text in a label

When editing your app, you need to select the control item so that the properties will appear on the right. To change the title that appears in the app, you first need to choose the label that has the text.

You can choose either of the following actions:

- Select **Tree view** and then scroll down to select **AppNameLabel1**.
- Select the title in the app and then **AppNameLabel1** will appear in the **Properties** pane.

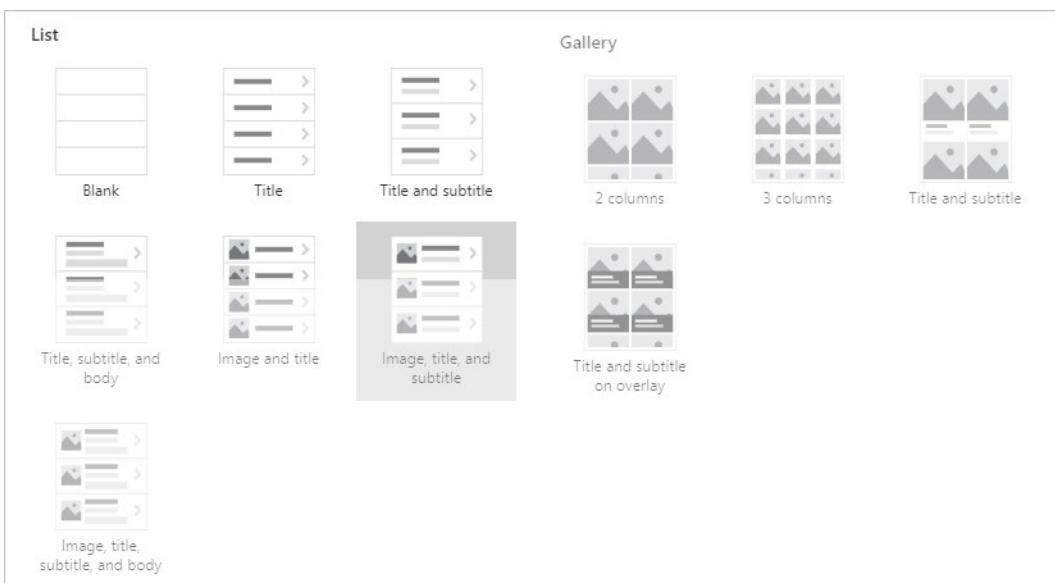
After the control item has been selected, you can edit the **Text** properties on the right side of the screen. You can also change other properties such as **Font**, **Font size**, and **Text alignment**. This process can be used to change the text and appearance of any label in the app.



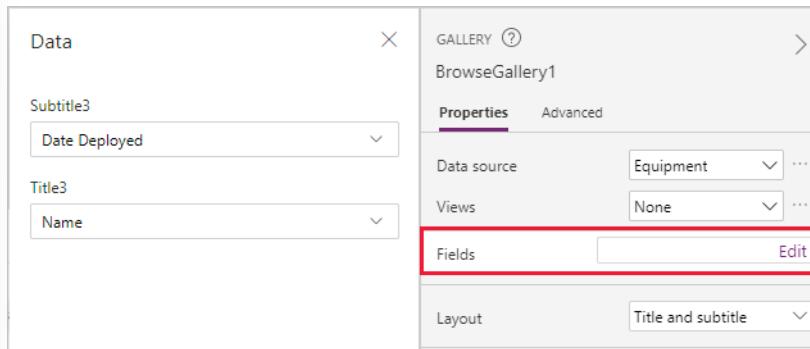
## Change the layout of a gallery

A **gallery** control displays multiple records for the user to view and select from. Each record in the gallery can display multiple types of information. For example, a gallery could show a listing of inventory items that show the name, model number, and price for each. Depending on what you need, galleries come in various vertical and horizontal layouts.

To change the layout of the gallery, select the gallery in the app or find **BrowseGallery1** in the tree view. Under the **Properties** pane, select from various **Layout** options.



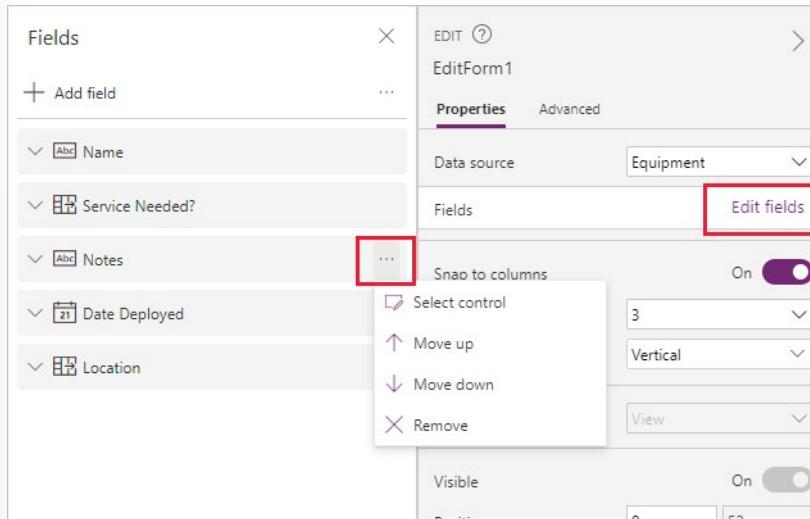
To change the content of the gallery's fields, go to the **Properties** pane and select **Edit** next to the **Fields** option. The areas that you can change will depend on the gallery layout that you selected. For each field, use the drop-down list to choose the data.



## Change the order in which fields appear

To change the order in which fields appear, follow these steps:

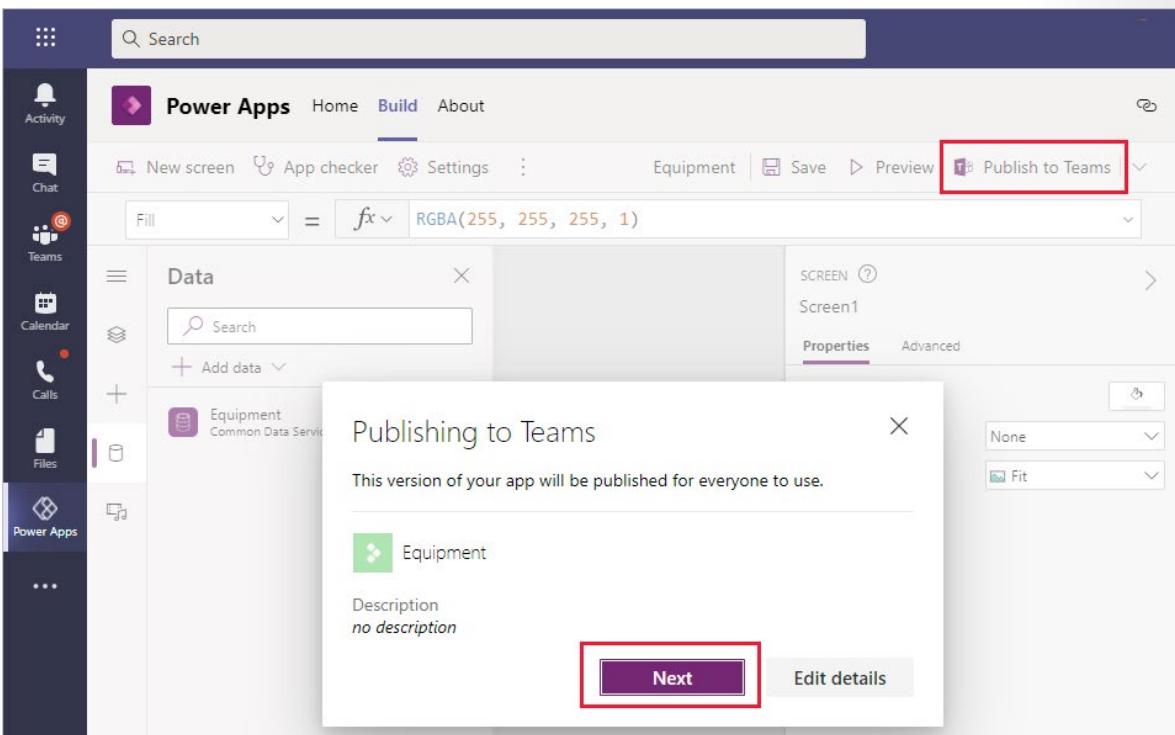
1. Select the form in the app or find **EditForm1** in the tree view.
2. Under the **Properties** pane, select **Edit fields** next to **Fields**.
3. Right-click the ellipsis next to each field to bring up more options. Select from **Move up**, **Move down**, or **Remove**.



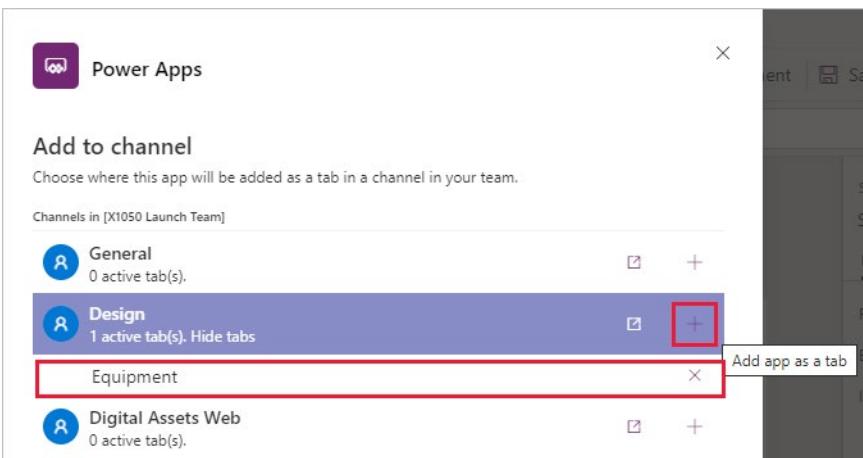
## Publish your app

Now that you have built an app, you will want to share it with your team. You can accomplish this task by publishing the app to Teams. After the app has been published, it will be accessible for your team to start taking advantage of it.

1. Open your app for editing in Power Apps Studio.
2. Select **Publish to Teams** from the toolbar and then select **Next**.



1. For each channel where you want the app to appear in a tab, select **Add app as a tab**. You can only publish to channels for the team that you created the app in.



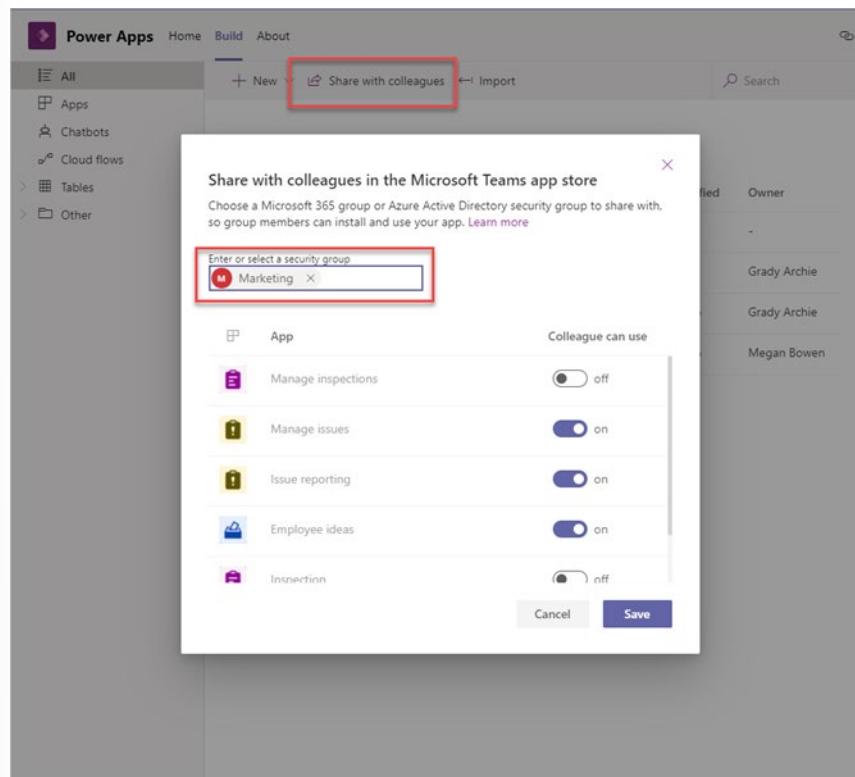
1. Select **Save and close**.

By default, your Dataverse for Teams app is only available to the team that you created the app in. Occasionally, you will want to share your application with others in your organization.

To share your app with others outside the team:

1. Verify that you are a team owner where the app is currently published.
2. Open the **Build** tab in the Power Apps for Teams app and then select **See all**.
3. Select **Share with colleagues**.

4. Enter the Microsoft 365 group or Microsoft Azure Active Directory (Azure AD) security group to share with in the **Enter or select a security group** field.
5. Turn on or off the sharing of apps to the users and then select **Save**.



Afterward, the app will appear in the **Built by your colleagues** page when users select **Apps** in Teams. To learn more about publishing your apps, see the links in the Summary unit at the end of this module.

## Install template apps

Microsoft created a few sample apps that can be used as-is or customized to suit your specific business needs. You can find these apps when you first open the app for Power Apps in Teams.

**Teams + Power Apps**

Microsoft Power Apps makes it easy to build apps for your business needs, from tracking time-off requests, managing inventory, submitting expenses, and more. [Learn more](#)

Build an app in 5 minutes

Create an app and connect it to your data to enhance collaboration across your team.

[Create an app](#)

Add these apps to your team

- Employee ideas**
- Inspection**
- Issue Reporting**

Each app has unique requirements and uses various Office 365 services. The install process might be slightly different depending on what is needed. To install any of these apps, left-click on it and then select **Add to a team**.

## Employee ideas app

With the **Employee ideas** app, you can collect ideas from your team and then review, manage, and vote on them to see which ones are worth exploring. Additionally, you can use campaigns to organize the team's ideas. The app will post each idea to a Team channel as they are submitted.

**Manage campaigns**

| Rank | Contributor | Ideas    |
|------|-------------|----------|
| 1    | Grady       | 10 ideas |
| 2    | Megan       | 3 ideas  |

| Rank | Idea               | Votes   |
|------|--------------------|---------|
| 1    | Company Masks      | 2 votes |
| 2    | In-Office Schedule | 1 vote  |

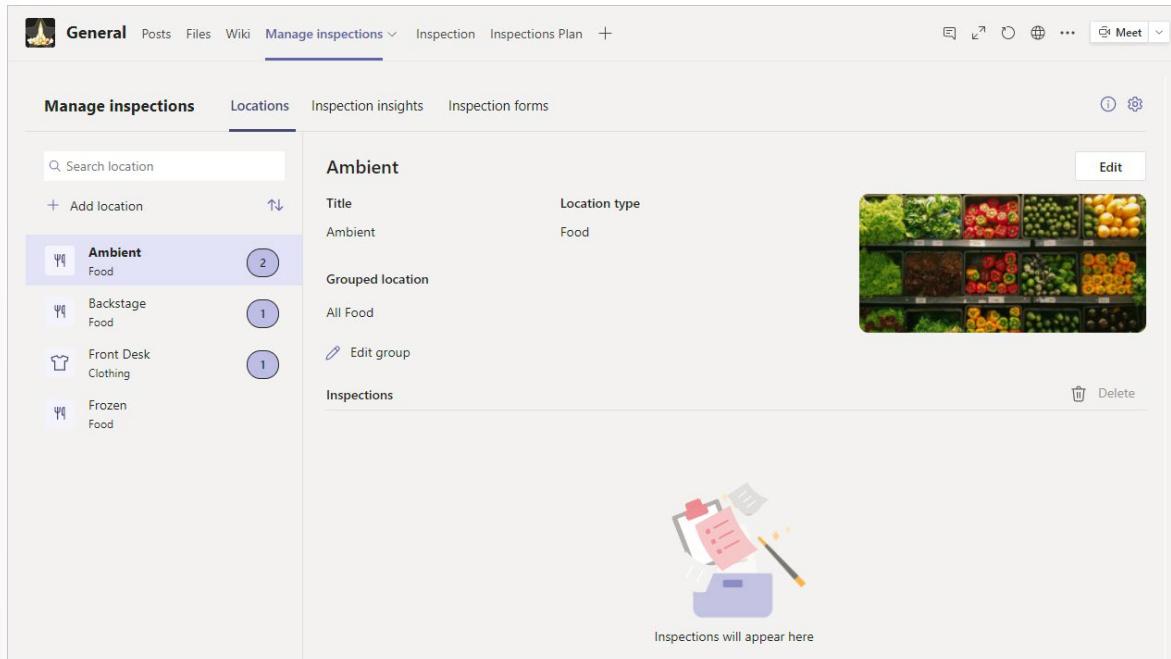
**Search campaigns** **Active** **Add campaign**

|                                                                                                                    |                                                                                                                      |                                                                                                                     |                                                                                                                                       |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| <b>Workplace Safety</b><br>We are looking for ideas to make our office a better workplace!<br>37 days left 2 ideas | <b>Café Food</b><br>We are looking for ideas to increase the food options and improve in food<br>50 days left 1 idea | <b>In-Person Meetings</b><br>Got any ideas for safer and more effective in-person meetings?<br>68 days left 2 ideas | <b>Return to Workplace</b><br>We are looking for ideas to help make our office safer for all of our employees<br>15 days left 6 ideas |
|--------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|

## Inspection app

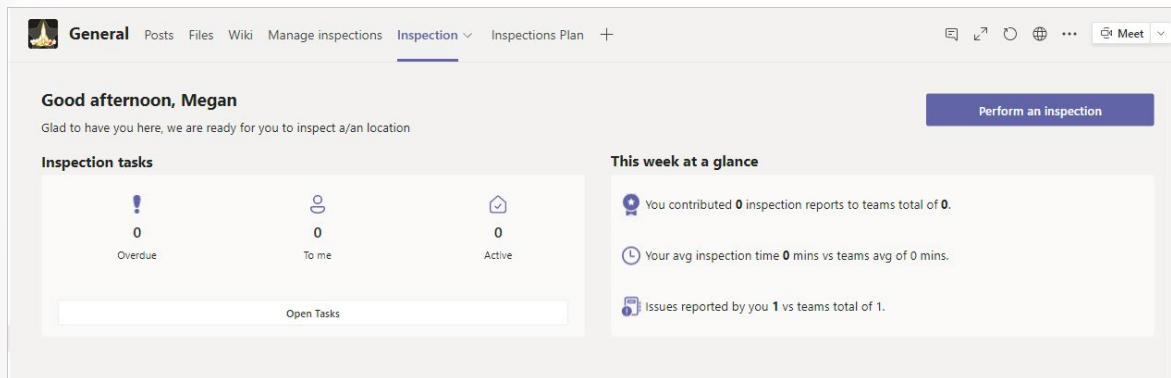
The **Inspection** app manages, schedules, and collects reports from your team. From this app, you can analyze the results. It uses Microsoft Planner to track the inspections. Two apps will be installed: **Manage inspections** and **Inspection**.

The **Manage inspections** app allows you to edit settings and define the type of inspections that your team can make. It also gives you an overview of everything that has been completed and will then provide reporting.



The screenshot shows the Microsoft Teams interface with the General channel selected. The top navigation bar includes Posts, Files, Wiki, Manage inspections, Inspection, Inspections Plan, and a Meet button. The Manage inspections tab is active. On the left, there's a sidebar titled 'Locations' with a search bar and a '+ Add location' button. Below it are four location groups: 'Ambient Food' (selected and highlighted in purple), 'Backstage Food' (1 item), 'Front Desk Clothing' (1 item), and 'Frozen Food'. The main content area displays the 'Ambient' location group with its title and location type. To the right is a placeholder image of a grocery store shelf filled with various fruits and vegetables. Below the image is an icon of a clipboard and pen, with the text 'Inspections will appear here'.

Each team member uses the Inspection app to view and update inspections that are assigned to them. Team members can start an inspection, review status, and track their own progress.



The screenshot shows the Microsoft Teams interface with the General channel selected. The top navigation bar includes Posts, Files, Wiki, Manage inspections, Inspection (selected and highlighted in blue), Inspections Plan, and a Meet button. The Inspection tab is active. The main content area starts with a greeting 'Good afternoon, Megan' and a message 'Glad to have you here, we are ready for you to inspect a/an location'. There is a large blue button labeled 'Perform an inspection'. Below the greeting, there are three sections: 'Inspection tasks' (Overdue: 0, To me: 0, Active: 0), 'This week at a glance' (You contributed 0 inspection reports to teams total of 0., Your avg inspection time 0 mins vs teams avg of 0 mins., Issues reported by you 1 vs teams total of 1.), and a 'Open Tasks' button.

## Issue reporting

Create a place for your team to report issues, assign tickets, and track progress with the **Issue reporting** app. This app uses Planner as a ticket system. Two apps are installed: **Manage issues** and **Issue reporting**.

The **Manage issues** app gives an overview of all tracked tasks and can create templates to help make it easier for your team to collect the needed information.

**Issue overview**

- 0 In progress
- 0 Not started
- 0 New issues today
- 0 Completed in past 7 days
- Average resolution time

**Recent issues**  
Recently submitted issues will appear here

**Overdue issues**  
Issues that are past due will appear here

**Issues by category**  
Issues submitted for each category will appear here

**Average resolution time by category**  
Average resolution time for each category will appear here

The **Issue reporting** app provides a personalized view for each team member, where they can view, edit, and create new items.

**Good afternoon, Megan.**

Glad to have you here, we are ready to help you report an issue.

**Issues reported by you**

| Completed | In progress | Not started |
|-----------|-------------|-------------|
| 0         | 0           | 0           |

**Last 7 days**

- The team has completed 0 issues
- The average completion time was 0 days
- There have been 0 issues reported

## Summary

Now, you can turn your great idea into an app for your team. This module showed you how to use Dataverse for Teams to build and publish apps from Power Apps. Data that is stored in Dataverse pro-

vides enhanced performance and scalability over other data sources. The hero template and the provided sample apps can help you build custom solutions to meet your organization's data and collaboration needs.

## Links that are related to modules for specific topics

For more information, see the following articles:

- Publish and share your app in Teams.
- Data types available in Dataverse.

## Module 3 Create a canvas app in Power Apps

### Get started with Power Apps

#### Introducing Power Apps

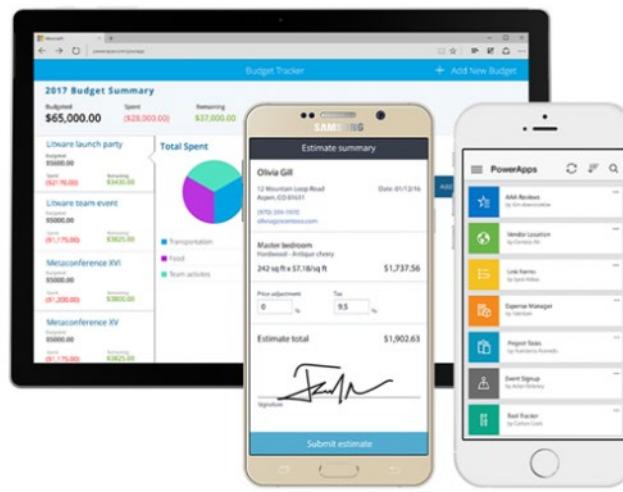
Power Apps is a suite of apps, services, connectors, and a data platform that provides you with an opportunity to build custom apps for your business needs. By using Power Apps, you can quickly build custom business apps that connect to your business data that is stored either in the underlying data platform (Microsoft Dataverse) or in various online and on-premises data sources (SharePoint, Excel, Office 365, Dynamics 365, SQL Server, and so on).

Apps that are built by using Power Apps provide rich business logic and workflow capabilities to transform your manual business processes to digital, automated processes. Power Apps simplifies the custom business app building experience by enabling users to build feature-rich apps without writing code.

Power Apps also provides an extensible platform that lets pro developers programmatically interact with data and metadata, apply business logic, create custom connectors, and integrate with external data.

With Power Apps, you can:

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



When it comes to using Power Apps to get things done and keep people informed, your options are nearly limitless. The following examples can help you think about how to use an app, instead of traditional paper notes, to run your business:

- **Equipment in the field** - Often, company representatives who visit customers in the field carry clipboards to help guarantee a paper trail of parts with scheduled replacement dates. By running an app on a tablet, reps can look up the customer's equipment, see a picture of a part, test and analyze the part, and then order new parts. Reps can perform these tasks on-site instead of leaving the customer's warehouse.
- **Restaurant employee management** - Employees of a large restaurant might fill out work schedules and vacation requests on a piece of paper that's affixed to a wall. With Power Apps running on everyone's smartphone, employees can open the app to record the same information, anywhere, anytime. The app can even send reminders for the start of the next day's shift.

If you're a beginner with Power Apps, this module gets you going quickly; if you're familiar with Power Apps, it ties concepts together and fills in the gaps.

## Power Apps building blocks

Power Apps is a collection of services, apps, and connectors that work together to let you do much more than just view your data. You can act on your data and update it anywhere and from any device.

To create, share, and administer apps, you'll use the following sites:

1. **Make a Power App<sup>1</sup>** - On this site, you can open apps, specify the type of app that you want to create, share your app, and create data connections and flows. To use this site, you'll need to log in by using your organizational account.
2. **Power Apps Studio<sup>2</sup>** - On this site, you build apps by configuring user interface (UI) elements and Excel-like formulas.
3. **Power Apps admin center<sup>3</sup>** - On this site, you'll define environments and data policies.

<sup>1</sup> <https://make.powerapps.com>

<sup>2</sup> <https://create.powerapps.com/studio/>

<sup>3</sup> <https://admin.powerplatform.microsoft.com/>

Note: To use these sites, you'll need to sign in by using your organizational account.

When you've completed your tasks, you can run your apps in a browser or in Power Apps Mobile (available for Windows tablets, iOS devices, and Android devices).

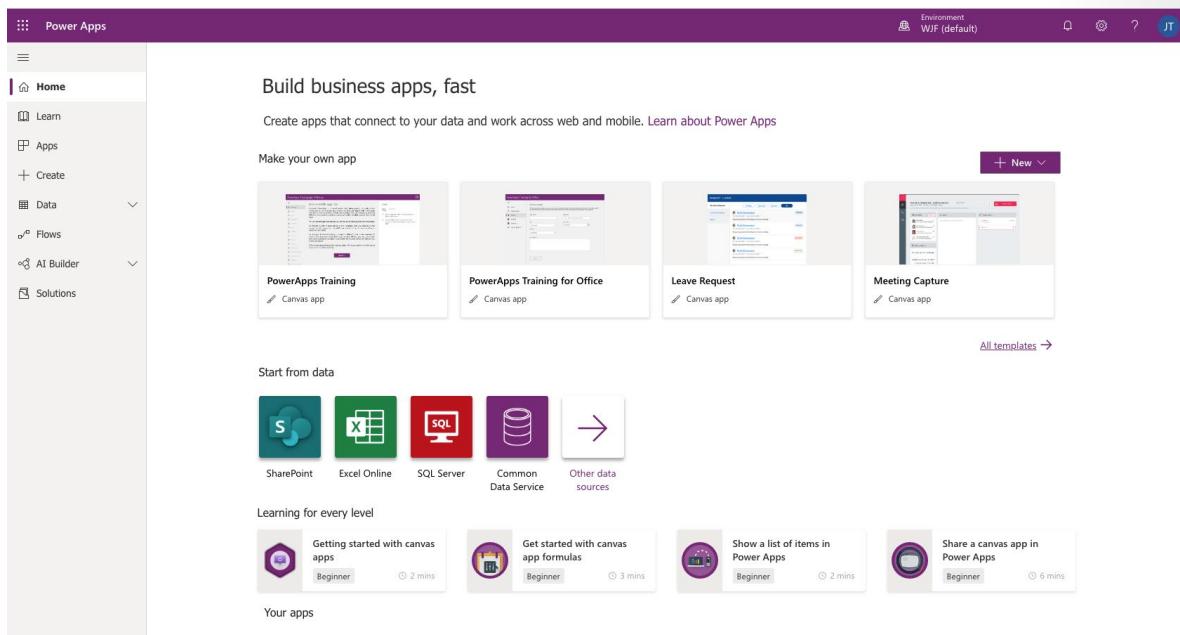
## Power Apps building blocks

This unit explores each part of the following Power Apps components:

- **Power Apps Home Page<sup>4</sup>** - Apps start here, whether you build them from data, a sample app, or a blank screen.
- **Power Apps Studio<sup>5</sup>** - Develop your apps further by connecting to data, adding and arranging user interface (UI) elements (known as controls), and building formulas.
- **Power Apps Mobile** - Run your apps on Microsoft Windows, Apple iOS, and Google Android devices.
- **Power Apps Admin Center<sup>6</sup>** - Manage Power Apps environments and other components.

## Power Apps Home Page

If you are building an app, you'll start with the **Power Apps Home Page<sup>7</sup>**. You can build apps from sample apps, templates, or a blank screen. All the apps that you've built appear here, along with any apps that others have created and shared with you.



**4** <https://make.powerapps.com>

**5** <https://create.powerapps.com/studio/?azure-portal=true>

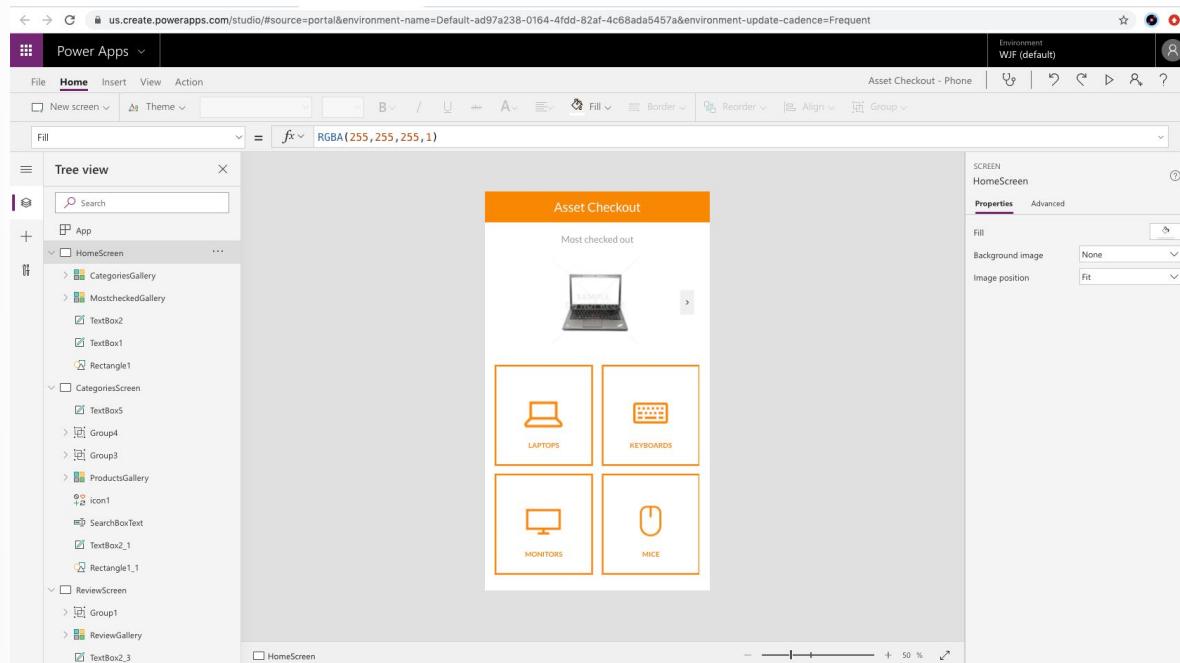
**6** <https://admin.powerplatform.microsoft.com>

**7** <https://make.powerapps.com>

## Power Apps Studio

Power Apps Studio is where you can fully develop your apps to make them more effective as a business tool and to make them more attractive. Power Apps Studio has three panes that make creating apps seem more like building a slide deck in Microsoft PowerPoint:

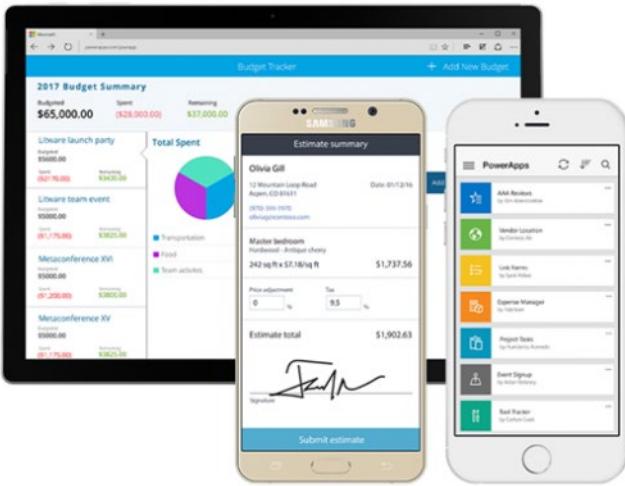
- **Left pane** - Shows a hierarchical view of all the controls on each screen or a thumbnail for each screen in your app.
- **Middle pane** - Shows the canvas app that you're working on.
- **Right pane** - Where you set options such as the layout, properties, and data sources for certain controls.



## Power Apps Mobile

Power Apps Mobile for Windows, iOS, and Android devices allows you to use all the apps that you've created, and those others have shared with you, on your mobile device. You or your users can download the Microsoft Power Apps app from the appropriate app store. When users log in with their credentials, they will see all apps that have been shared with them. The Power Apps Mobile app only needs to be downloaded once.

When you use apps in Power Apps Mobile, you get the most out of your device's capabilities: camera controls, GPS location, and more.



## Power Apps Admin center

The Power Apps admin center is the centralized place for managing Power Apps for an organization. On this site, you can define and manage different environments to house the apps. For example, you might have separate environments for development and production apps. Additionally, you can define data connections and manage environment roles and data policies.

## Licensing

Most users get their initial start with Power Apps by utilizing one of the licenses that comes with their Microsoft 365 Plan or Microsoft Dynamics 365 license. These licenses allow you to extend the functionality of the app that is licensed. This means if you purchased a Microsoft 365 plan that included a Power Apps license then you can build apps that extend and use SharePoint as a data source. But Power Apps doesn't have to stop at just extending that platform.

Power Apps has over 300 available data source connectors available including Dataverse. To incorporate Dataverse or any of those additional connectors all users of the app will need a premium license. There are two different ways to acquire a Premium license:

- Per App model
- Per User model.

The Per App license allows users to access premium connectors for a specific business solution. The Per User license allows users to run unlimited premium licensed apps. This gives you the ability to grow with Power Apps and control cost by purchasing the license that best matches your business goals.

In addition, Power Apps also has the capability to use Power Apps portals to build externally or internally facing websites using Dataverse and Power Apps controls. Power Apps portals have their own licensing model and are not included in any of the licenses discussed previously above. With Power Portals you will purchase a capacity based license to meet your business needs.

Review the following links about licensing.

### Microsoft Power Apps pricing<sup>8</sup>

<sup>8</sup> <https://powerapps.microsoft.com/pricing/?azureportal=true>

**Microsoft Power Automate pricing<sup>9</sup>**

**Microsoft Power Apps portals pricing<sup>10</sup>**

## Power Apps related technologies

Microsoft Power Apps works with other technologies to help you build powerful apps for your organization. Some of these technologies include:

- **Data sources** - Without data, you don't have a business. Data sources bring cloud and on-premises data into your apps. You access data through built-in connections, custom connectors, and gateways.
- **Dataverse** - A compliant and scalable data service that's integrated into Power Apps.
- **Power Automate** - Allows you to build automated workflows to receive notifications, run processes, collect data, and more.

## Data sources, connections, and gateways

In Power Apps, most canvas apps use external information that is stored in Data Sources. A common example is a table in an Excel file that is stored in OneDrive for Business or SharePoint. Apps access these data sources by using connections. Some connections allow Power Apps to read and write stored data. In Power Apps, you can add many data sources to your apps through built-in or custom connectors. Some of the most popular data sources are shown in the following figure.

|                                                                                     |                     |                                                                                     |                       |
|-------------------------------------------------------------------------------------|---------------------|-------------------------------------------------------------------------------------|-----------------------|
|   | Common Data Service |   | Office 365 Outlook    |
|  | SharePoint          |  | Excel                 |
|  | SQL Server          |  | OneDrive for Business |
|  | Dynamics 365        |  | OneDrive              |
|  | Office 365 Users    |  | Dropbox               |

Many data sources are cloud services, like Salesforce. Even Twitter can be a data source if, for example, you're tracking your company's hashtags. Connectors might not seem like the most exciting part of app development; however, they're essential when you work with data that you, your colleagues, and your customers care about. When an app shows up with your data source for the first time, you might suddenly find that they are, in fact, exciting.

For data that's stored on-premises instead of in the cloud, you can use a gateway to provide a reliable connection between Power Apps and your data source. The gateway sits on an on-premises computer and communicates with Power Apps.

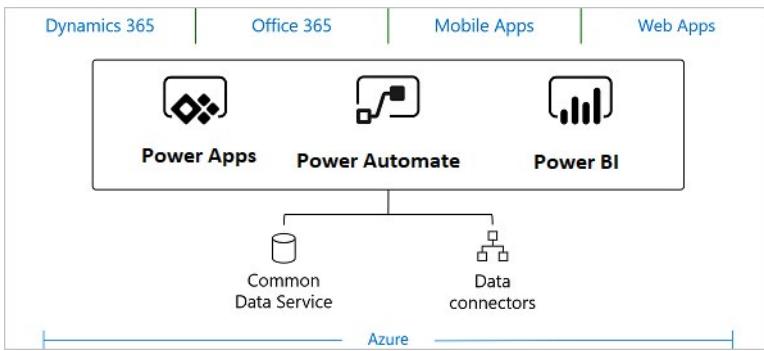
An advantage of building your business apps in Power Apps is being able to connect to many data sources in a single app. With the connectors in Power Apps, you can connect to where your data lives. To learn more about data sources in Power Apps, refer to the Working With Data learning path.

<sup>9</sup> <https://us.flow.microsoft.com/pricing/?azureportal=true>

<sup>10</sup> <https://powerapps.microsoft.com/portals/?azureportal=true>

## Dataverse

An important data source option to explore further is the Dataverse. Dataverse lets you store and manage data that's used by business applications. Data within Dataverse is stored within a set of tables. A table is a set of rows that are used to store data, similar to how a table stores data within a database. Dataverse includes a base set of standard tables that cover typical scenarios, but you can also create custom tables that are specific to your organization and then populate them with data by using Power Query. App makers can then use Power Apps to build rich applications by using this data.



For information on purchasing a plan to use Dataverse, refer to the [License<sup>11</sup>](#) and [Pricing<sup>12</sup>](#) information pages.

## Reasons to use Dataverse

Standard and custom tables within Dataverse provide a cloud-based storage option for your data. tables let you create a business-focused definition of your organization's data for use within apps. If you're unsure if tables are your best option, consider the following benefits:

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

## Related Power Platform technologies

As you continue developing your application, you may want to consider implementing additional Power Apps related technologies such as Power Automate and or Power BI. For example, you may have a simple Expense Report App that requires an approval before an item can be purchased. With Power Automate,

<sup>11</sup> <https://docs.microsoft.com/power-platform/admin/pricing-billing-skus>

<sup>12</sup> <https://powerapps.microsoft.com/pricing/>

you can create a simple Flow to make this happen. Or maybe you want to display your data with custom charts and graphs giving your users a more visual look into the data, which can often be useful. In this section, you will learn more about some of the other Power Platform technologies and how you could apply them in your own Power Apps solution. Keep in mind, if you decide to implement these Power Apps related technologies you should also review their licensing structure and associated costs.

## Powerful alone. Better together.



Power Apps



Power BI

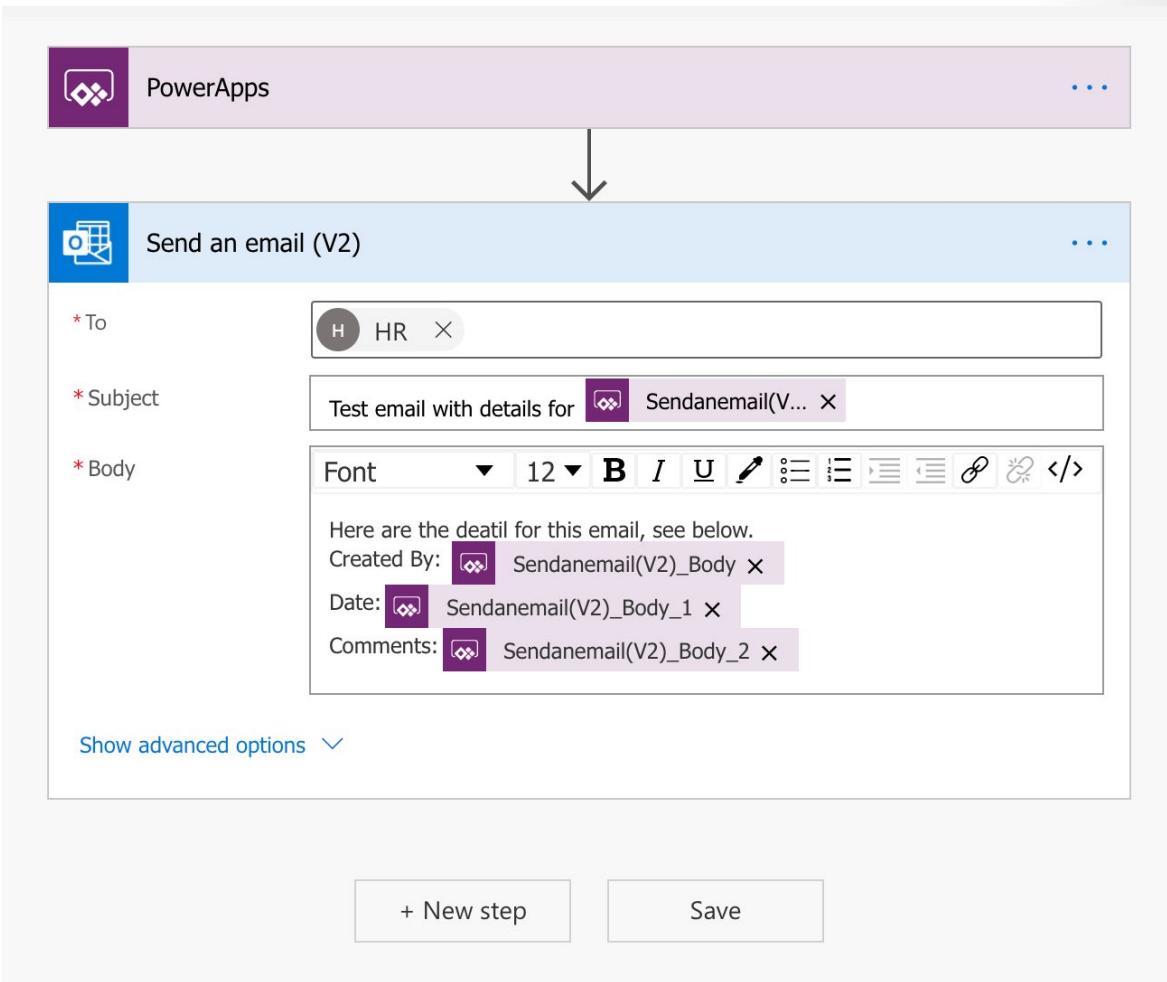


Power Automate

### Power Automate

Power Automate brings automation to your business. This can be traditional workflows via flow, Robotic Process Automation (RPA) for automating legacy systems via UI Flows, or business process automation via Business Process Flows. Each of these capabilities increases your productivity to connect disjointed systems to build the business solution you need and make your app more powerful.

You can use Power Automate to create logic that performs one or more tasks when an event occurs in a canvas app. For example, configure a button to execute a flow to do one of the following: create an item in a SharePoint list, send an email or meeting request, or add a file to OneDrive. The button could be configured to do all of those in a single Power Automate flow. You can configure any control in the app to start the flow, which continues to run even if you close Power Apps. Below is an example using Power Automate to send a flow:



## Identify Flows in your Solution

Now that you have a general overview of Power Automate, how do you determine if the solution you're building requires a Flow? There are a number of simple functions Power Apps can do, like sending an email when a button is pressed in your application. This email generated from Power Apps can also contain dynamic/specific information and be sent to any email address you would like. Often, customers will use Power Automate to create this same functionality even though Power Apps can do this out of the box. Power Automate should be used for more complex solutions, such as the approval workflows. With Power Automate you can run an approval when a button is pressed, on a schedule, when an item is created or modified, and so on.

For many Power Apps solutions Power Automate is used to handle complex business logic. Do you need a way to make sure someone actioned on the incident report that was generated by your app? Or, do you need a process to kick off every time new data is created in another system so Power Apps will have the data it needs? Do you need to check each morning to see if an inspection is due that day and then send an email with a link to your Power Apps inspection form? These are great uses of Power Automate to transform your app from a point solution to a fully featured business solution.

## Power BI

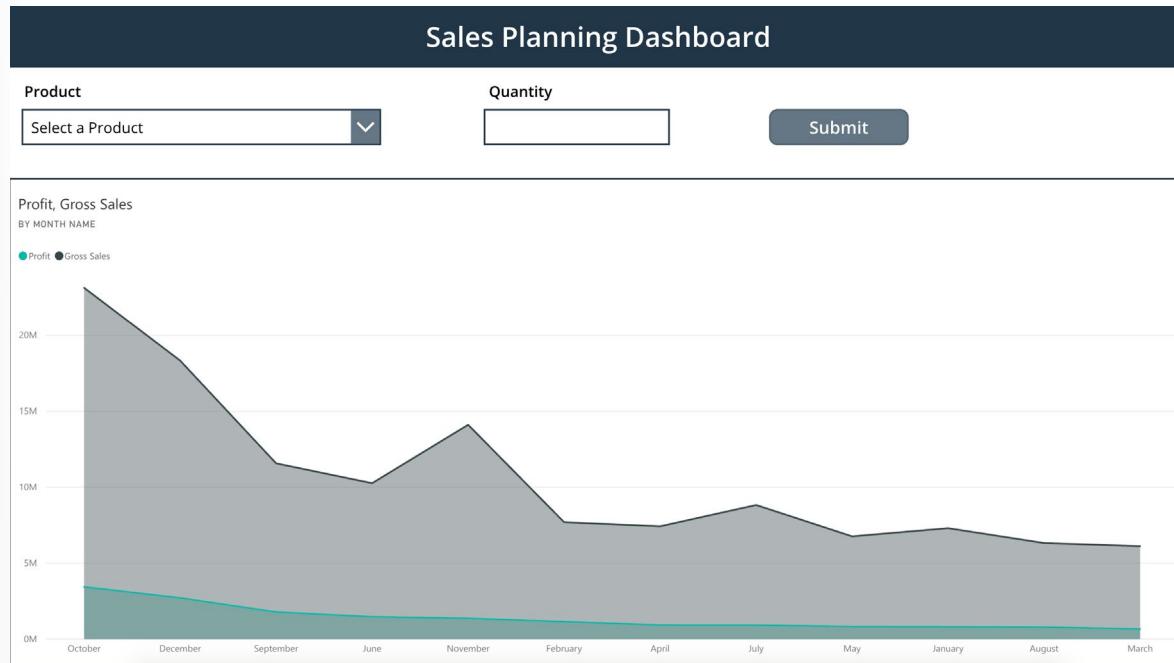
Power BI is an analytics tool within the Power Platform suite. Power BI connects data from multiple sources and transforms the data into graphical visualizations to gain insights. It allows business users to utilize a number of different visualizations to build comprehensive reports and dashboards. When creating Power BI reports to view and analyze your app data, you have the ability to customize them for personal use and will only be accessible by you, providing you with a more unique and custom experience. If you need to share the report with others, you and each of the report consumer will need a Power BI Pro license. This license allows you to not only share the content but also control what others are able to do with the shared report or dashboard.

While Power Apps has capabilities to include simple graphs or tables, many solutions would be better served with a visualization provided by Power BI. Power Apps and Power BI have two options for seamless integration:

### Embed a Power BI tile in a Power Apps app

By embedding a Power BI tile in a Power Apps solution, you are able to bring valuable visualizations into the app to allow the user to consume that data within the context of the app.

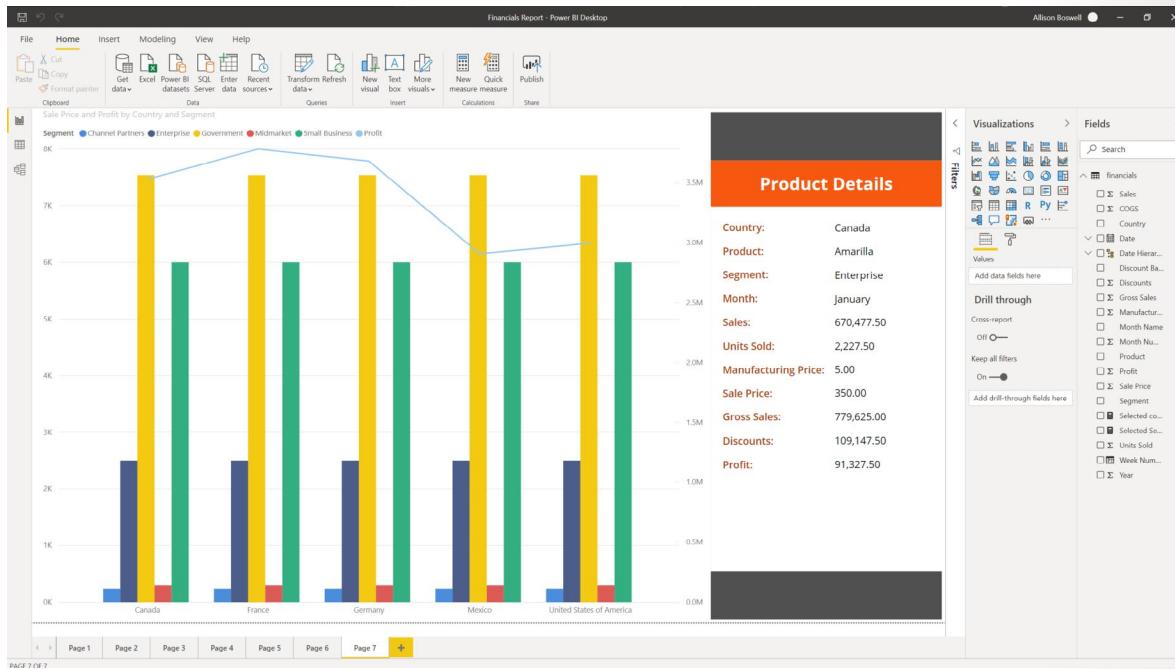
In the example below, you will see a simple Power BI Tile embedded in a Sales Planning app built in Power Apps. The visual is displaying the Profit and Gross Sales and the Power Apps form allows the user to enter sales predictions.



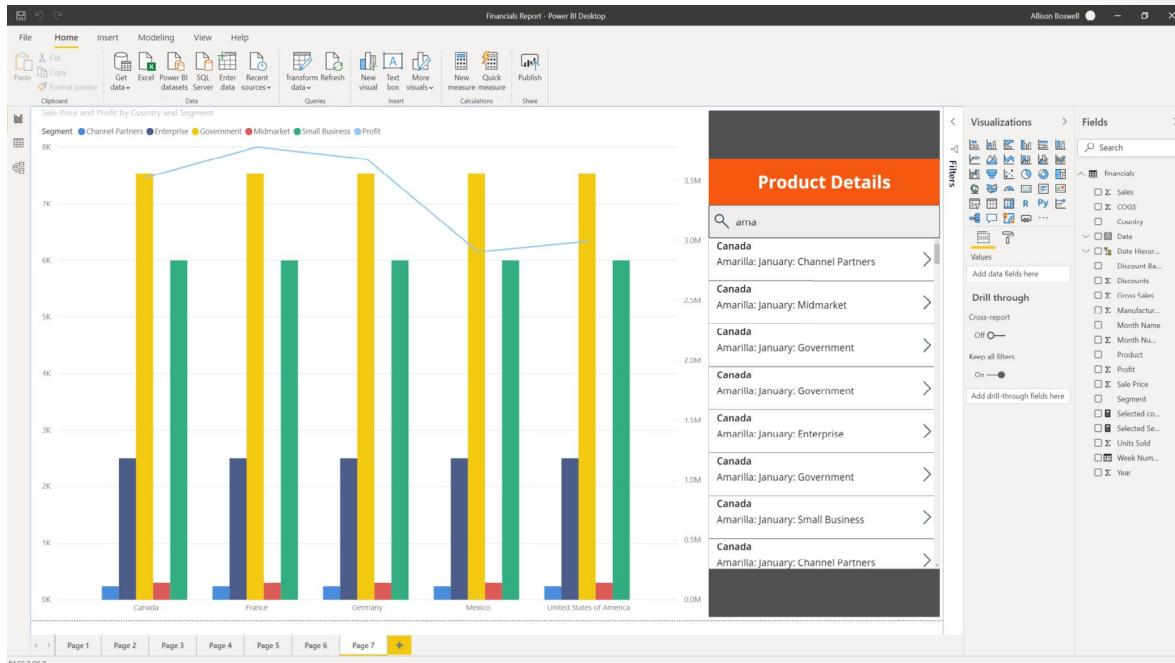
### Embed a Power Apps app in a Power BI Dashboard

Another integration between these two applications, is to embed a Power Apps app in your Power BI report. This allows the user to action on data while never leaving the dashboard resulting in a better user experience. Consider an inventory management dashboard for a manufacturing facility. Without leaving the dashboard, the user can submit to purchasing an order for additional material. While the solution may have been utilizing both the Power Apps and Power BI platforms, the user simply experiences a complete end to end solution in one window on their desktop.

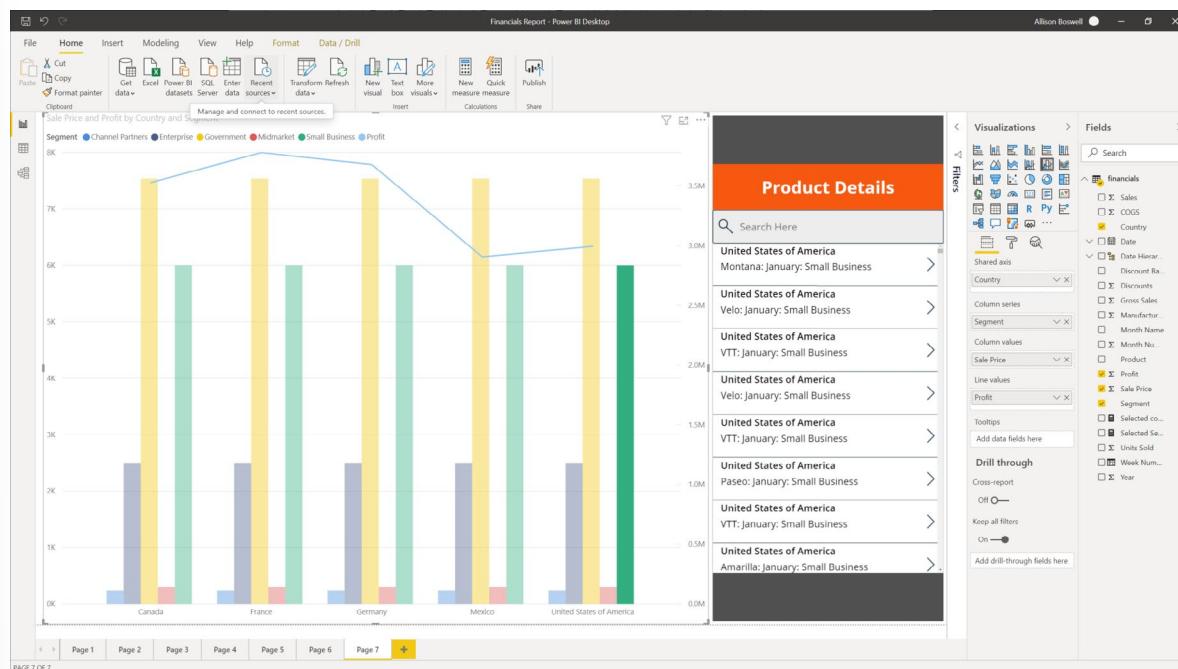
In the example below, we are analyzing the Sale Price and Profit by Country and Segment. Notice once you have embedded your Power App in a Power BI Dashboard you can navigate between screens.



In this next screenshot, still working with the same data as the previous example, you can utilize the native Power Apps features like Search with Power BI data.



In this last screenshot, for this example, you will see the embedded Power App is filtered by the Power BI selection.



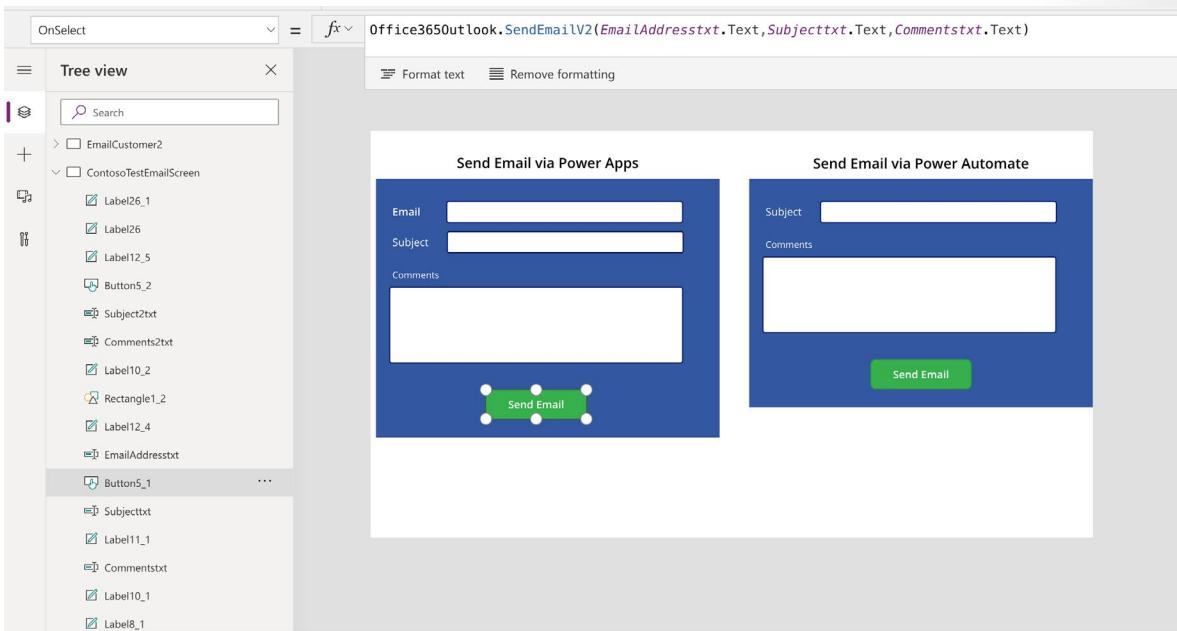
## Translating needs to the appropriate technology

To build the best solution, think through the use cases and determine how you want to collect the data, use the data, and analyze the data. Once you have determined how the solution will be used in each one of those cases, you can begin to select the right technology to execute each function.

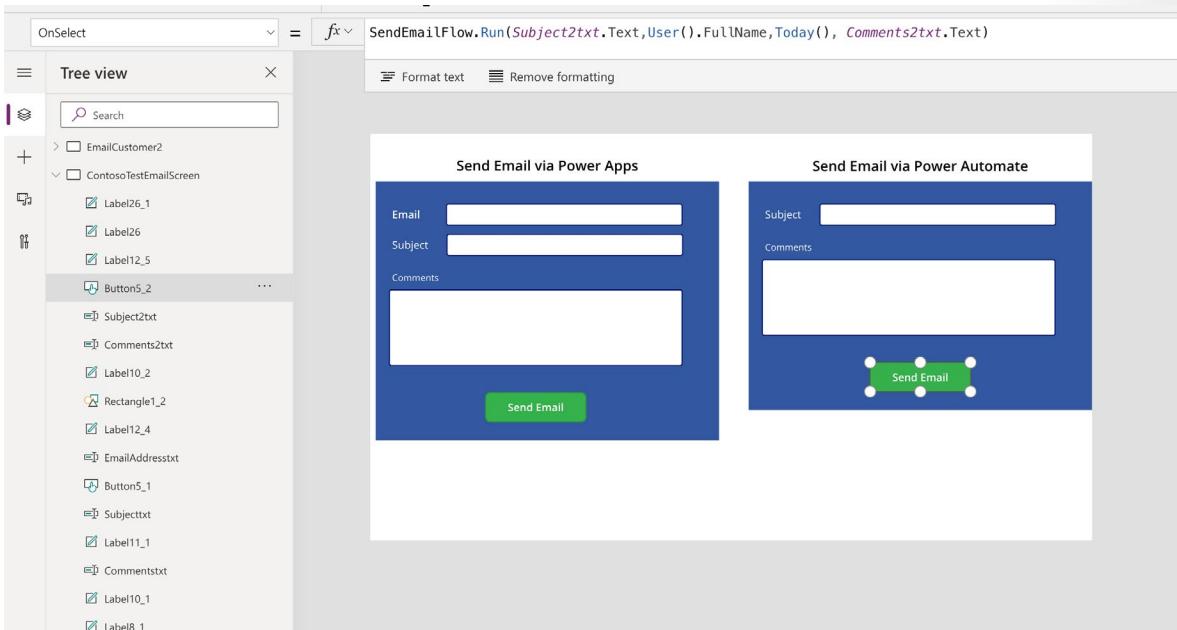
It would be difficult to cover every use case and decision point, but to help you understand the decision-making process let's explore sending an email via Power Automate versus sending an email via Power Apps. First consider the look and feel of the email, does your solution require special formatting of the email? To format the text of your email in Power Apps, like adding italics or bold text, you would need to write HTML. In Power Automate though, this functionality can be implemented by using the simple Design Interface that is provided out of the box.

Below are examples of the formulas to execute sending an email via Power Apps versus via Power Automate.

### Send an Email via Power Apps



### Send an Email via Power Automate



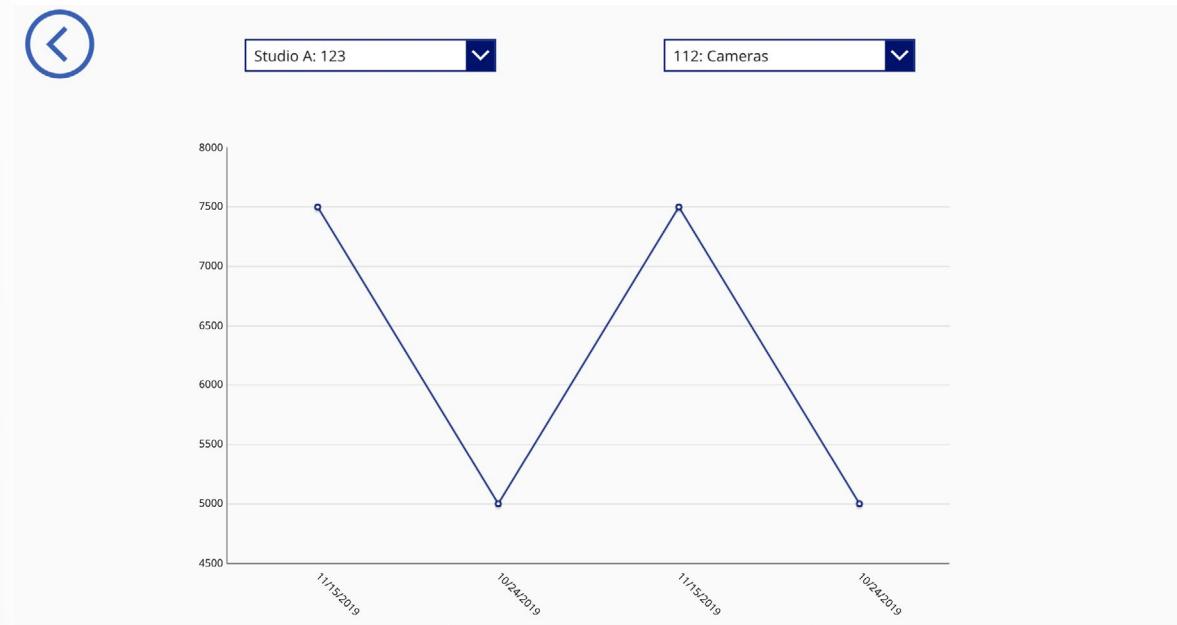
Also, the number of steps in your solution/process will aid you in determining which technology best suits your needs. Power Apps is great for performing simple solutions with minimal steps but as your solutions become more complex and requires multiple steps, Power Automate would be the better solution.

Again for this particular example, both technologies can provide the same solution, but there are little nuances that should be considered and thoroughly discussed during the design process to determine your requirements and help you choose the best product for your solution.

Let's not forget about discussing Power Apps and Power BI, and when to use one vs. the other. When deciding whether to use the basic charts, graphs, and visuals that come with Power Apps out of the box

or to utilize a more powerful software like Power BI it really depends on your business solution and requirements. For example, if in your solution, you are wanting to add some basic graphs and charts to improve the apps overall look and feel while adding some visual flair for your users, Power Apps has you covered.

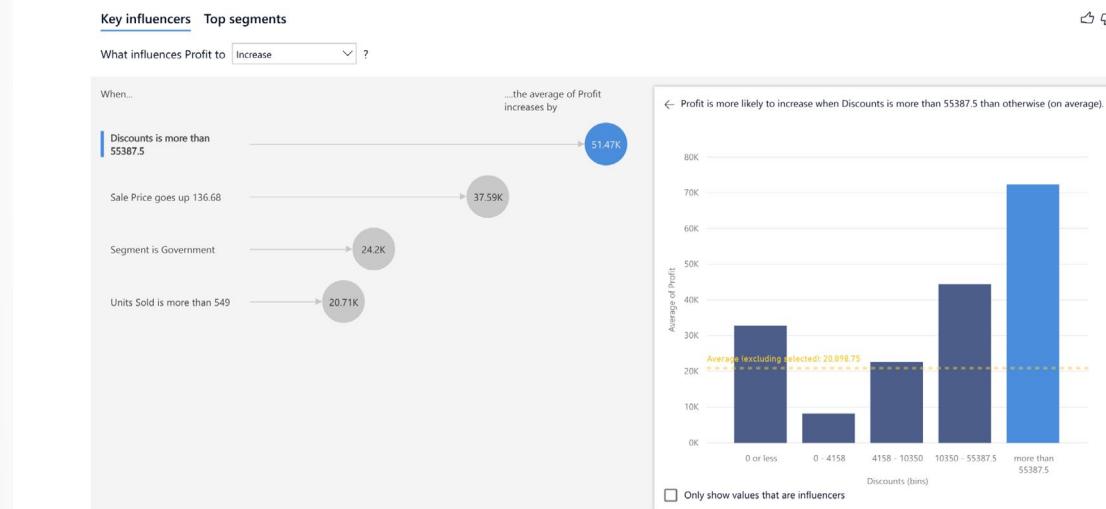
Here is a quick look at one of the simple, out of the box Power Apps charts.



Simple and minimal design above, nothing fancy but it gets the job done.

On the other hand, if your solution requires in-depth analysis of your data, and robust visuals, Power BI will be the best product for your solution. Keep in mind, with Power BI, each app user will need an additional license on top of the Power Apps license. This is a small price to pay though if our solution relies on intuitive dashboards, charts, graphs, and several other features to help you get the most out of your solution.

Financials Report  
PAGE 1



By identifying the needs of related Power Apps technologies in your solution and strategically implementing them, you will be able to provide your users with a better overall experience when using the solution.

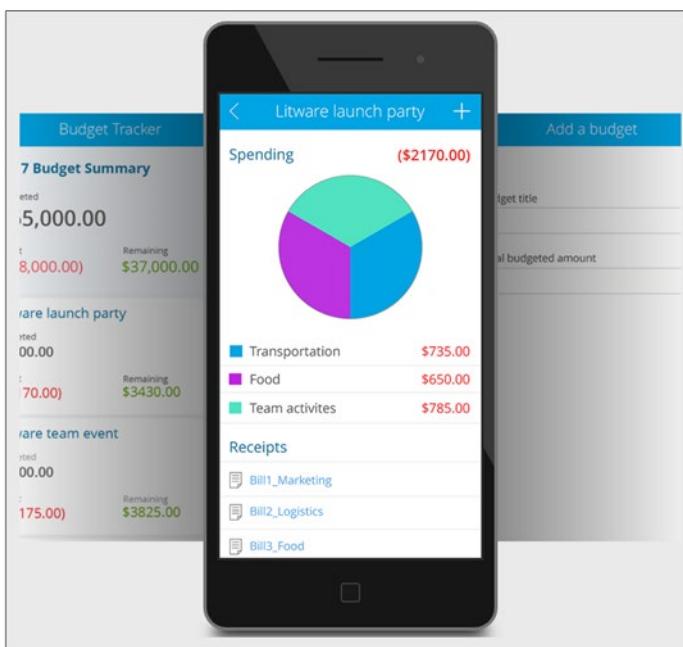
## Ways to build Power Apps

This unit describes how to create an app from a template, a blank canvas, and a data source. In the next unit, you'll be able to get more hands-on with app creation. This learning path is focused on canvas apps, which give you the flexibility to arrange the user experience and interface the way that you want it. You can get started in many different ways; however, for all of the options, you will use the Power Apps Studio features and functionality to build your app.

### Create an app from a template

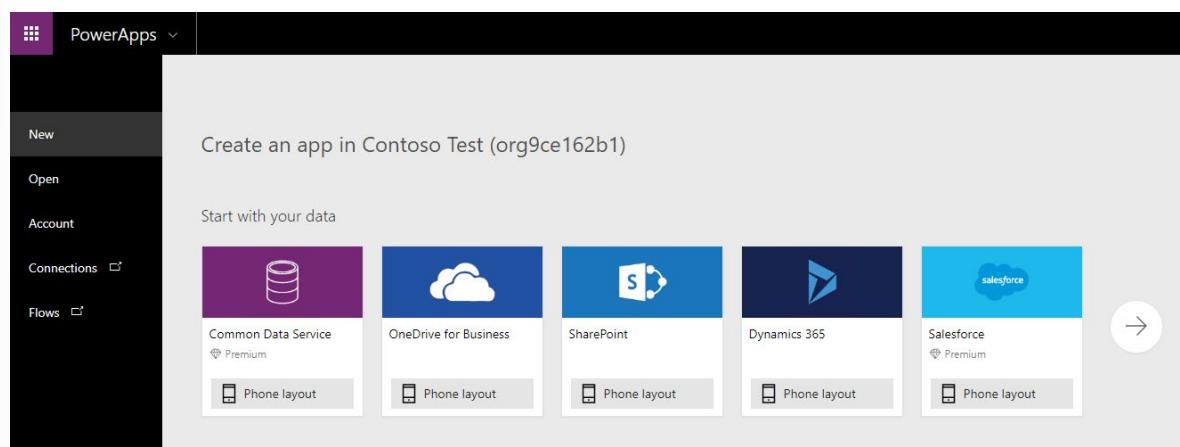
A good way to create an app is to start from a template. Templates use sample data to help you determine what's possible. By opening templates in Power Apps Studio, you can learn, hands-on, how an app is built.

For example, you can use the Budget Tracker template to create an app that helps you track the budget for projects and events with custom categories, simple data entry, and visuals that highlight expenditures for an effortless inspection.



### Create an app from a data source

Another great way to get started is to generate an app from your own data. Simply point Power Apps at the data source of your choice (for example, a list in Microsoft SharePoint or Dataverse), and watch as Power Apps automatically builds a three-screen app. This three-screen app lets you display, edit, delete, and create rows.



A special data source is SharePoint. Modern lists in SharePoint and Power Apps have a tight integration. You can either build an app from within a SharePoint site or you can use Power Apps to customize your modern list forms.

A screenshot of a SharePoint list titled "Clients". The ribbon bar at the top has tabs for "Office 365" and "SharePoint". In the ribbon, there is a "PowerApps" button with a dropdown menu containing "Create an app" and "Customize forms". The list itself shows columns for "Name", "Category", and "Image", with several items listed such as "Ageless Beauty Clay", "Bianco Carrara Marble", and "Golden Teak".

The following app was created from a SharePoint list and lets you browse items in the list, view item details, and create and edit items. After Power Apps generates an app, you can customize it to make it look and behave exactly the way you want.

A screenshot of a PowerApp interface for a "FlooringEstimates" list. On the left, there is a list view showing items like "Ageless Beauty Clay", "Bianco Carrara Marble", and "Golden Teak". On the right, there is a detailed view of a selected item, "Brazilian Koa". The detailed view includes fields for "Name" (set to "Brazilian Koa"), "Category" (set to "Hardwood"), and "Image" (a preview of a wooden floor sample). Below the image is an "Overview" section with the text: "Its distinctive orange coloring and brown/black striping make it one of the most unique and exotic species available on the market today." There is also a "Price" field set to "5.69".

## Build from a blank canvas

You can also build an app from the ground up and add all the pieces as you go. You can then branch out and use your imagination.

## Designing a Power Apps app

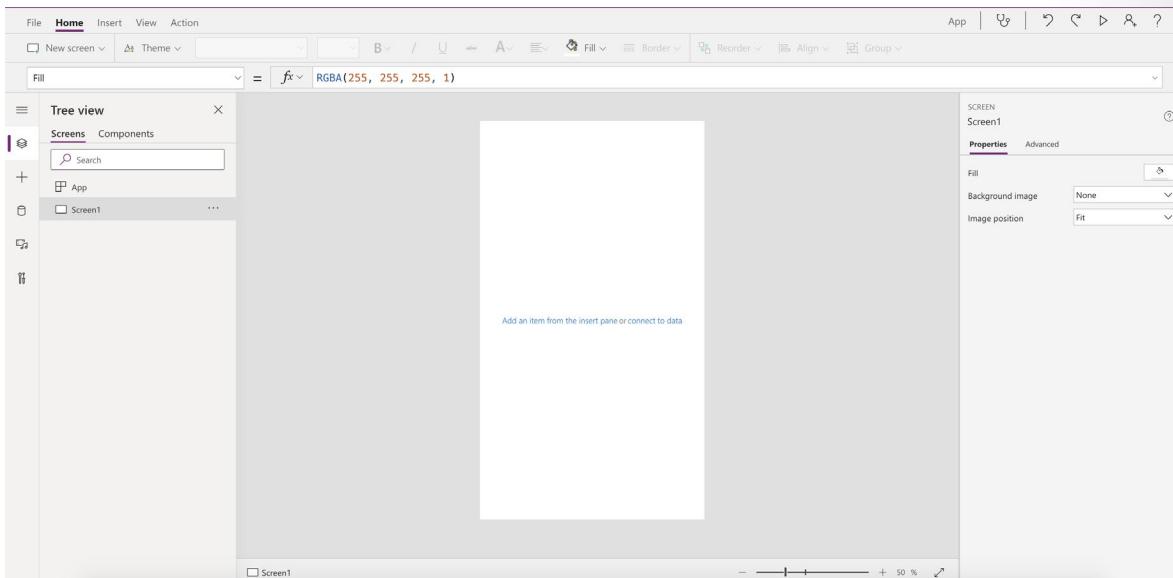
As an App Maker, before you begin building your Power Apps solution, it's recommended to go through a design process.

When designing your Power Apps solution, there are several different factors to consider, such as:

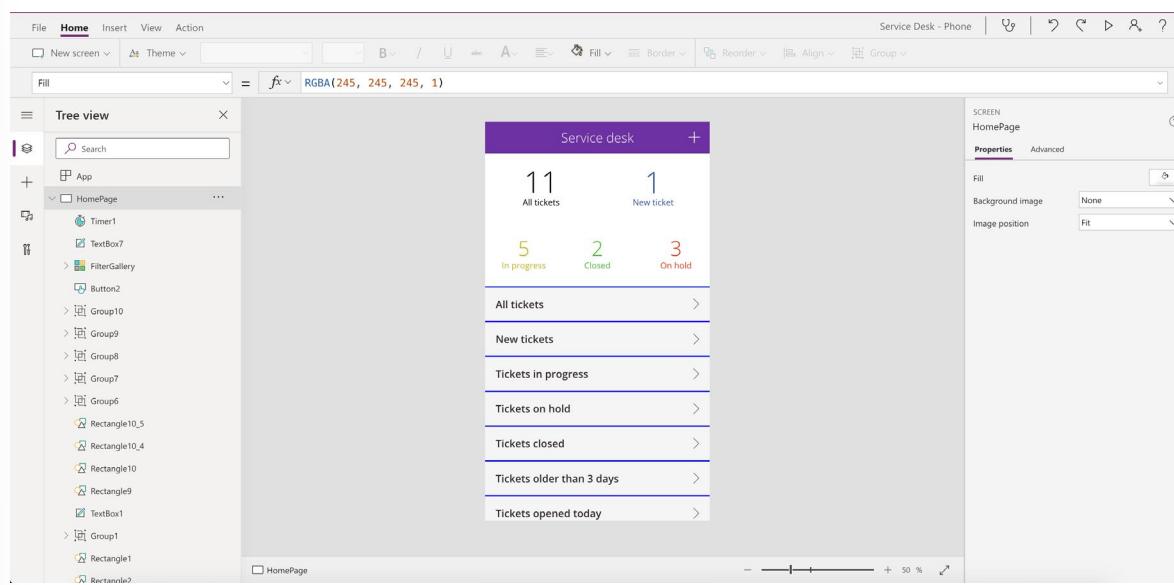
- Business requirements
- Data Model
- User Experience (UX)
- User Interface (UI)
- Business Logic
- Output

By going through a simple design process, you can flush out any minor issues before they become a larger problem once the app has been put into production. It is also important to understand that this design process is for Canvas apps.

So how do you go from a simple blank Canvas app, as seen below?



To a fully customized Canvas app solution?



## Understand the needs of the user

One of the most powerful and, at the same time, challenging parts of building a canvas app is that you start with a blank screen. This gives you the ability to build what you want, but to do that you have to know what you want.

In many cases, when purchasing software to solve and or streamline business solutions, there are many business processes that don't quite fall within the software's supported guidelines. When you run into this issue, typically, there are several internal discussions and meetings held to determine how those unsupported processes can be updated/ altered to meet the software requirements. For most organizations, this isn't ideal because of the cost or time takes to update those business processes. The great news is, by using Power Apps to build your solution, you won't have to worry about unsupported business solutions. Why? With Power Apps, you can build a custom solution tailored to the exact needs of your business requirements.

Often when building an app, you are tempted to recreate the piece of paper or legacy software-driven process exactly. This is possible but might not be the ideal solution. By challenging the existing process and asking what it is the business needs to do, not what does the piece of paper or old software allows you to do, it opens the possibility of better, more efficient processes. For example, maybe on the paper process, the user had to type notes about what they see. Would it be better instead to just take a picture? This type of thinking will lead to better apps and better outcomes.

## Business Requirements

Every app you develop will have a different set of business requirements based on the solution. Taking the time to think about all the requirements is key to rolling out a successful production app.

Depending on the solution or company policies, you may have certain security, privacy, or compliance requirements you must follow. For example, let's say you are collecting secure personal information in the app. You will want to ensure this information is securely stored and not visible to everyone.

During this process, you will also want to identify any government regulations or authentication/authorization requirements (if applicable). You don't necessarily have to have all the answers to your questions here; you just want to know all the requirements.

## Offline Mode

One of the first questions to consider when developing your application is, will the app need to function offline? If so, will the entire app or only part of the app needs to function offline? When will the data be synchronized to my data source? Are there any limitations?

This is important to consider during the planning phase because if you were to build your app without this functionality, then decide to add it later, it will be more difficult than just doing it in the first place. Why is this? You will need to make sure you are using collections and additional functions like SaveData and LoadData as you go along to allow your app to function offline. Also, if you are using Forms and trying to implement Offline mode, you will run into issues.

There is a thorough discussion that needs to take place around Offline mode, and it's best to have this early in the design process as it will affect the rest of the process.

## Data Model

In the "Power Apps related technologies" module, you learned about some of the common data sources for building apps, but with all these choices how do you actually decide which data source to use for your solution? Maybe you already have a data source implemented that users work with on a day to day basis, like SharePoint. Could you just use this as your data source to build your app? Do I need to connect to multiple data sources? These are all common questions you should ask yourself and there are number of additional factors to consider, such as:

- **Business Requirements** – Every data source and its supported functionality is slightly different. So, depending on your app requirements you need to select the data source that supports your needs or modify your business requirements to comply with the supported functionality for the selected data source.
- **Licensing/Cost** – Certain data sources like the Dataverse or SQL are considered a "premium data source". A premium data source will require each user who uses the app to have a Power Apps Per App Plan or a Power Apps Per User Plan. For more information about licensing, see **Power Apps pricing**<sup>13</sup>

## User Experience (UX)

By designing your Power Apps solution in a Canvas app, you have complete control of the end-user experience. This allows you to fully customize nearly every aspect of your app. However, just because you can doesn't necessarily mean you should. When designing your Power Apps solution your goal should be to keep it simple. When your end users open the application and begin using it, they should have no confusion about what to click on or where to go. If your app requires an extensive training program for users to understand how to use it, you may want to rethink your app.

Some of the basic designs elements you will want to consider are things like:

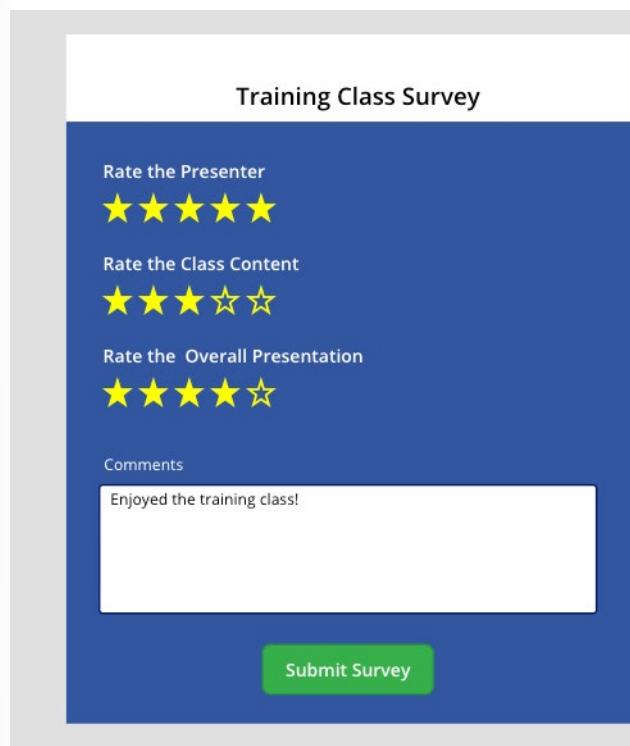
- Custom Branding (your logo and colors)
- Pop-ups
- Hide/show buttons based on users' access/permissions

One of the most common User Experience enhancements you can implement in your applications are Pop-ups. By implementing pop-ups, you can provide the users with a simple, but useful visual to confirm what the user clicked on went through or maybe your pop-up acts as a loading screen as the logic on the

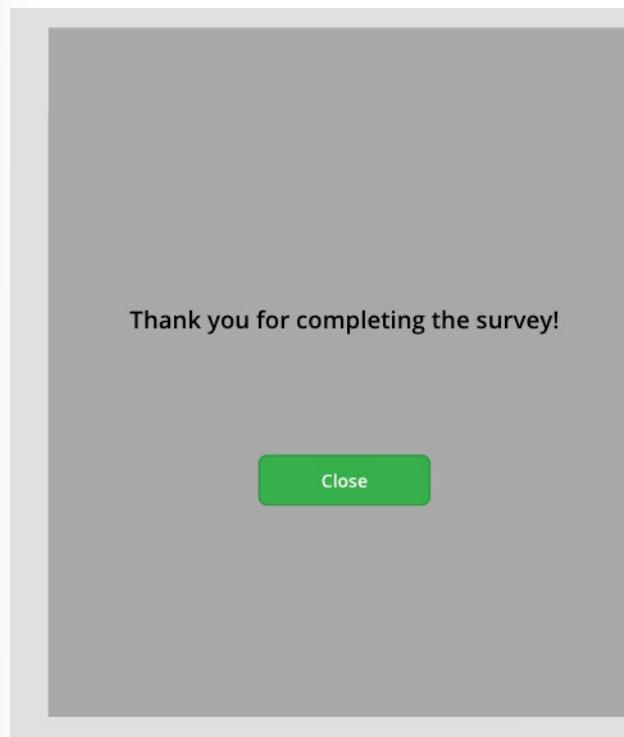
<sup>13</sup> <https://powerapps.microsoft.com/pricing/?azureportal=true>

backend is processed. For example, in the screenshot below when a user clicks on "Submit", we might have a simple pop-up display to let them know their submission was successful.

In this example app, the user completes a Survey for the training class they just attended.



Once all of the information has been written to the data source successfully, a pop-up is generated to confirm the submission was successful.



Without confirmation, a user may not be sure if their submission was accepted. They may try to click the button again, causing incorrect or inconsistent data being written to the data source.

Remember, these are not the only customizations you can make to the app, these are just some of the common ones. Another thing to keep in mind as you add different design features is the more logic you add for the customization of the app the more code your application will need to process. So, for example, if you add several different functions for hiding buttons, or showing popups on a given screen, this could cause your application performance to slow as each piece of code runs.

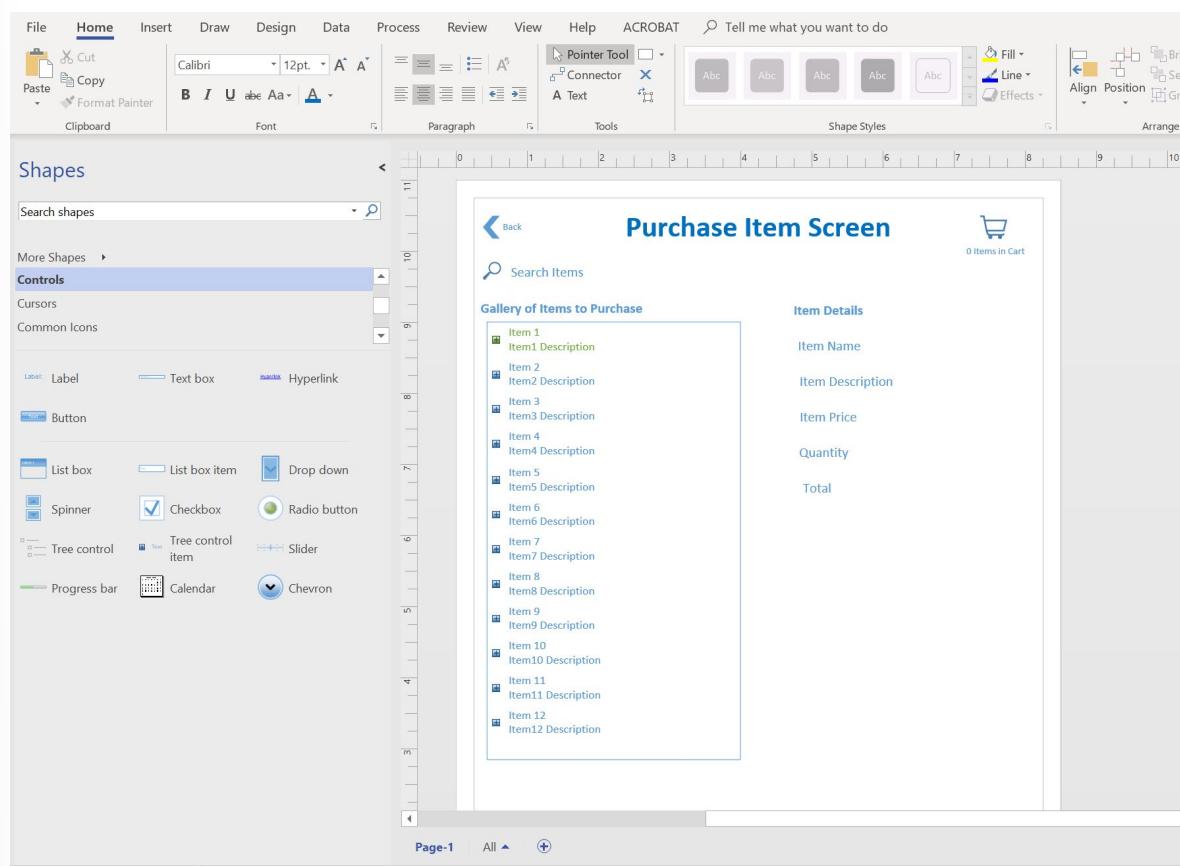
Finally, challenge yourself to do better with your user experience. Maybe today, the user records room temperatures by clicking in a box, changing the device keyboard from letters to numbers, and then typing in "70". A better option may be to replace with a slider control that defaults to 70 and ranges from 65 to 75. Then, with a swipe of their finger, they can record the temperature. Small changes like this make for happier, more productive app users.

## User Interface (UI)

To fully visualize the User Interface or UI, you may want to consider creating a mockup of your application. Two common ways to create a mockup of your application are below:

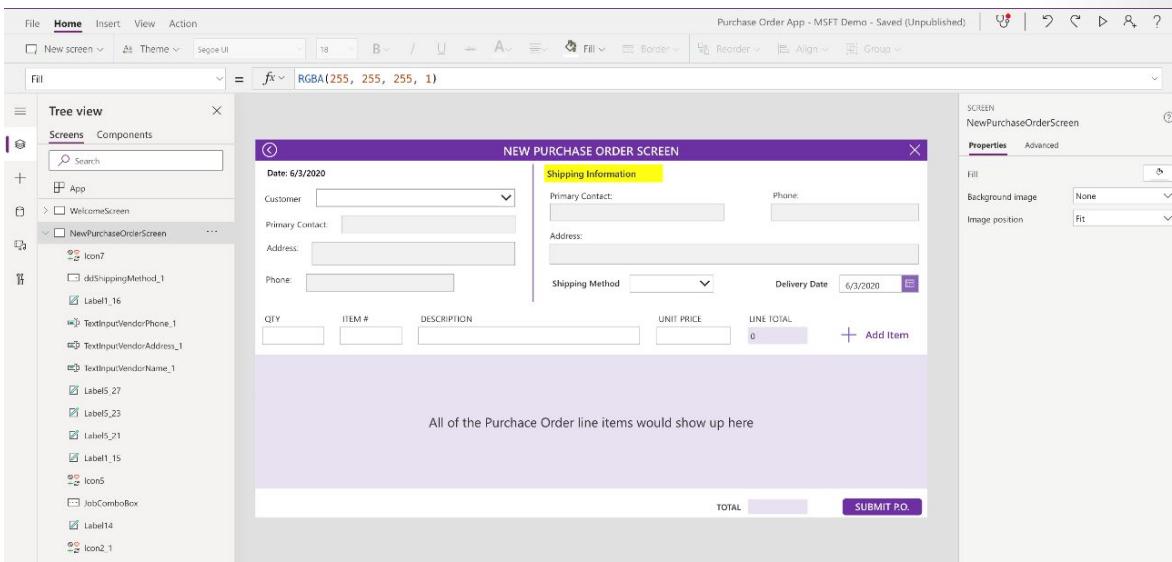
- Use Visio to create a wireframe diagram. A wireframe is a visual representation of an application's user interface. To begin, there are various website and mobile wireframe templates available, or you could start from blank template. The diagrams are a quick way to show app functionality and gain team consensus on the design.

The example below shows a simple Visio wireframe of a Purchase Items screen in an inventory app.



- Use Power Apps to create a mockup of your application. You can add most of the controls, graphics, forms, and other items to your app screens and play with the layout and size for each element as if you were building the app for real. When designing the UI you don't need to add the logic behind the various elements you placed on the screen. The goal here is to focus on what the app could look like and how it could function. This is similar to what you can do with a Visio wireframe but one of the biggest pros of going this route is that you will gain more experience working with Power Apps and also learn more about the various UI elements available in the process. All of the experience and knowledge you will gain by creating your app mockup in Power Apps will only payoff later when it's time to start on the production app. Another big upside to using Power Apps for your mockup is that if you show this to your team and they like what you did, you can continue building off this app or create a new app and copy the elements you would like to keep to your other application. By not having to redo the UI or only having to redo parts of it, you could potentially save yourself hours of work.

The example below shows a simple mockup of a New Purchase Order Screen.



It really comes down to your preference and comfort with the software you are using to create the mockup. You should also consider licensing and costs when making this decision. Visio requires additional licensing to get the full functionality required for creating a wireframe diagram. Whereas with Power Apps, it doesn't matter which license you have. As long as you have Power Apps (and sufficient permissions in your environment), you can create apps and mockup apps.

As you design the User Interface, a few additional things to think about are Accessibility and Localization. It's important to ensure the app interface follows accessibility guidelines so all your users can interact with your application without any issues. To review these guidelines and additional accessibility properties, see [Create accessible canvas apps in Power Apps<sup>14</sup>](#).

Localization can be something you must consider when developing your application as well. Depending on where your app will be used, you may need to use different punctuation. For example, some regions of the world use a . (dot or period) as the decimal separator while others use a , (comma). For more information on building a globally supported application, see [Build global support into canvas apps<sup>15</sup>](#).

## Business Logic

When using the Dataverse, you can create business rules and recommendations to apply logic and validations without writing code or creating plug-ins. The great thing about the Dataverse and business rules is that they are applied at the data level. This means that you can apply rules that are enforced regardless of how the data is accessed.

Often when building apps all of the business logic is built into the app. This works great if the data is only accessed via the app. The challenge is often business data is used in many ways and from different tools. This is where Business Rules shine. You can apply logic on the data in the Dataverse, allowing your rules to be enforced no matter which tool interacts with the data.

For example, you have built a capital project expense tracking application using Dataverse as the data source. In your business process, the duration column is an optional column if your request is less than 10,000 but the duration column is required if the request is more than 10,000. After you set up your table in Dataverse, you would then apply a business rule that says if Project Amount is greater than 10,000 then

<sup>14</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/accessible-apps?azureportal=true>

<sup>15</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/global-apps?azure-portal=true>

make Project Duration a required column. Now, regardless of how the user interacts with the data, the Business Rule will be enforced, keeping your data integrity.

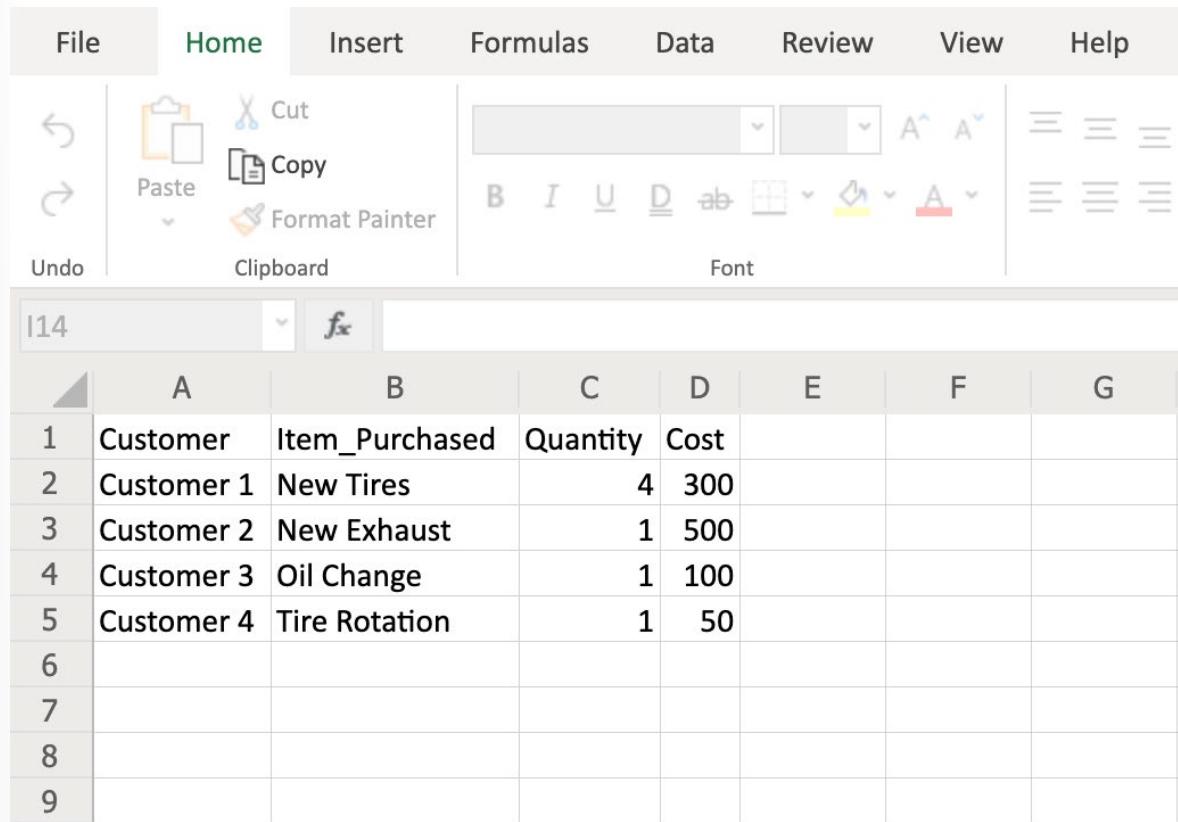
## Output

Finally, you will want to discuss your app's data output. This simply means what type of data will your app generate, and once the data is generated what will be done with it? A few questions to ask your app stakeholders:

- How does the data need to be visualized?
- What actions will be taken on the data once it is collected?
- Are there specific format or file types the data are needed?

The answers to these questions will help determine if additional functionality needs to be added to the app such as a Power BI report, email output, PDF, or CSV.

Let's look at an example. Perhaps your organization has a legacy ERP solution and the orders submitted in your Power App need to be reflected in the ERP application. While one option might be to build a custom connector to that solution, another option may be to export the data to a CSV file using Power Automate and Power Apps together, see screenshot below:



A screenshot of the Microsoft Excel ribbon interface. The ribbon tabs shown are File, Home, Insert, Formulas, Data, Review, View, and Help. The Home tab is selected. Below the ribbon is the Excel toolbar with icons for Undo, Redo, Paste, Cut, Copy, Format Painter, and other common functions. To the right of the toolbar is the Font section of the ribbon. The main area shows a table of data in a grid. The table has columns labeled A through G. Row 1 contains the headers: Customer, Item\_Purchased, Quantity, and Cost. Rows 2 through 5 contain data: Customer 1 (New Tires), Customer 2 (New Exhaust), Customer 3 (Oil Change), and Customer 4 (Tire Rotation). The Quantity and Cost values are aligned in columns D and E respectively. Row 6 is blank, followed by rows 7, 8, and 9.

|   | A          | B              | C        | D    | E   | F | G |
|---|------------|----------------|----------|------|-----|---|---|
| 1 | Customer   | Item_Purchased | Quantity | Cost |     |   |   |
| 2 | Customer 1 | New Tires      |          | 4    | 300 |   |   |
| 3 | Customer 2 | New Exhaust    |          | 1    | 500 |   |   |
| 4 | Customer 3 | Oil Change     |          | 1    | 100 |   |   |
| 5 | Customer 4 | Tire Rotation  |          | 1    | 50  |   |   |
| 6 |            |                |          |      |     |   |   |
| 7 |            |                |          |      |     |   |   |
| 8 |            |                |          |      |     |   |   |
| 9 |            |                |          |      |     |   |   |

The great thing about generating this CSV file export is that it's not linked to your data, so the changes you make to the file will not alter the app data.

For more information, see [Planning a Power Apps project<sup>16</sup>](https://docs.microsoft.com/powerapps/guidance/planning/introduction).

<sup>16</sup> <https://docs.microsoft.com/powerapps/guidance/planning/introduction>

## Exercise - Create your first app in Power Apps

In this unit, you'll generate a mobile app where the data source is a Microsoft Excel workbook that's stored in Microsoft OneDrive for Business. This Excel workbook lists a company's inventory of flooring samples with pictures and prices.

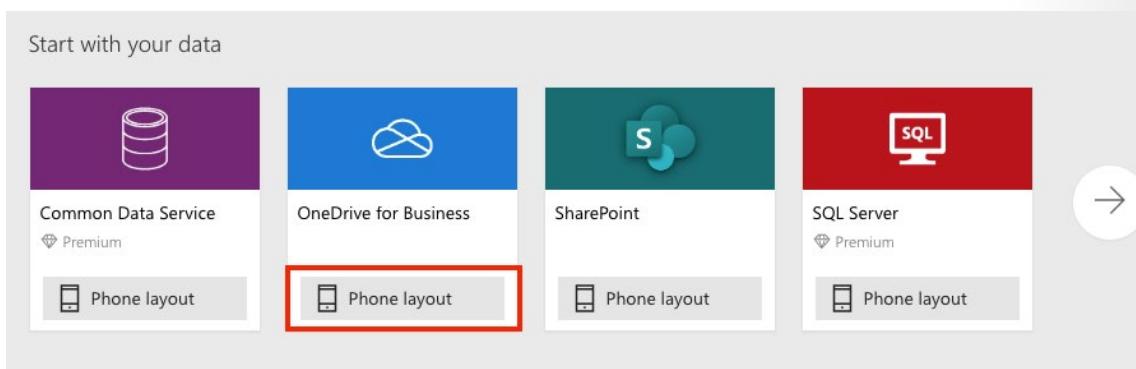
Keep in mind that you can use data from many other sources, including Microsoft SharePoint, cloud services like Salesforce, and on-premises sources like Microsoft SQL Server.

*Note:* Power Apps requires either an Office 365 license or a free trial. Learn more about your licensing options. **Microsoft products include Microsoft Power Apps and Power Automate.**<sup>17</sup>

### Connect to a data source

To connect to a data source, use the following procedure:

1. Download the **Flooring Estimates workbook**<sup>18</sup> and save it to OneDrive for Business.
2. Go to <https://make.powerapps.com> and sign in with your organizational account.
3. In the left pane, select **Apps**.
4. Select **+ New app** and then **Canvas** from the drop-down menu.
5. For the **OneDrive for Business** data source, select **Phone layout**.



Generated apps are always based on a single list or table, but you can add more data to the app later. The next three steps explain how to connect to the Excel workbook.

6. Under **OneDrive for Business**, select **Create**.
7. Under **Connections**, select **OneDrive for Business** and browse to the file location.  
You might need to select **New Connection** to see the **OneDrive for Business** connection.
8. Under **Choose an Excel file**, select the **FlooringEstimates.xlsx** file.
9. Under **Choose a table**, select the **FlooringEstimates** table.
10. Select **Connect** on the bottom right.

Power Apps generates the app by inspecting your data and matching it with Power Apps capabilities so that you get a working app as a starting point.

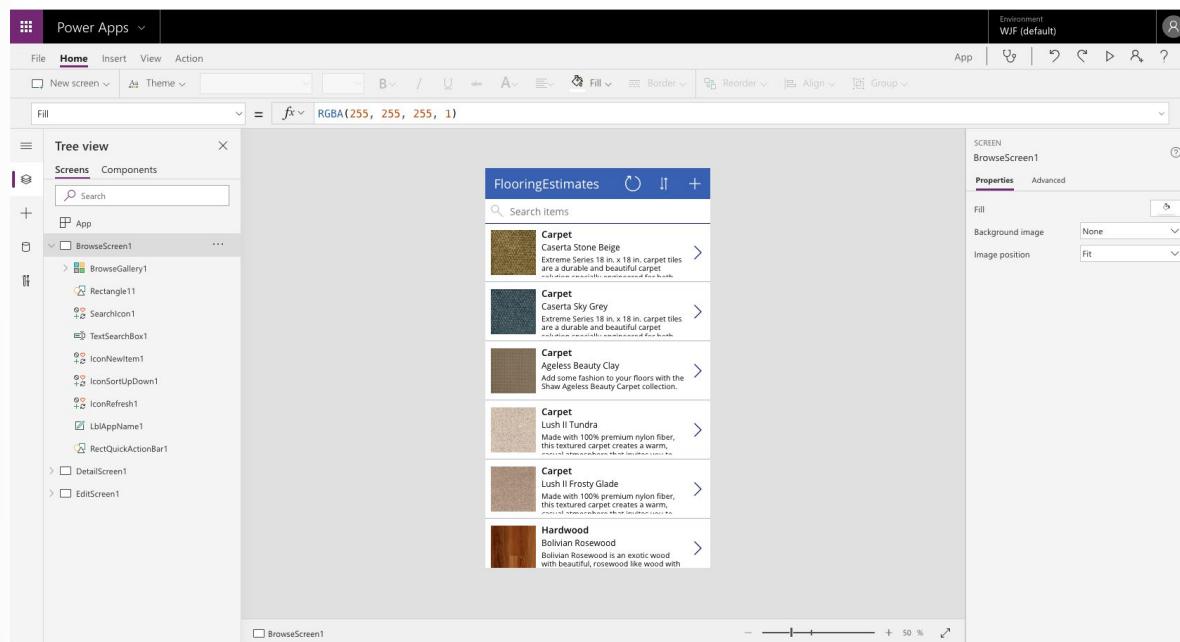
<sup>17</sup> <https://docs.microsoft.com/powerapps/administrator/pricing-billing-skus>

<sup>18</sup> <https://az787822.vo.msecnd.net/documentation/get-started-from-data/FlooringEstimates.xlsx>

## Explore the generated app

Your new three-screen app now opens in Power Apps Studio.

The following figure shows the main development window for Power Apps Studio, which you'll learn more about in later units.



Select **Play** in the upper-right corner to practice using the app. Notice that it includes all the data from the table and provides a good default experience.

All apps that are generated from data have the same set of screens that you can view from the Screens pane:

- **Browse screen** - This screen appears by default. In it, you can browse, sort, filter, and refresh the data from the data source. In the browse screen, you can add items to the data source by selecting the plus sign (+).
- **Details screen** - The details screen shows all information about a single item. In this screen, you can open an item to edit or delete it.
- **Edit/create screen** - In this screen, you can edit an existing item or create a new one.

To make your app visible on the phone, it needs to be saved. Select **File**, **Save as**. Replace the current title "App" with **flooring-estimates app**, and then select **Save**. You will see a green check mark when all changes are successfully saved. You can now open the app on your phone.

## Install the app on your device

To see how the app runs on mobile, install the Power Apps Mobile app on your phone. When building an app, you should test it in the same form factor as your users.

1. Download Power Apps Mobile from the app store for the platform that you want to use.
2. Sign in by using your username and password.

3. On your phone or tablet, run the flooring-estimates app in Power Apps Mobile. If you do not want to install the app, you can run it in a browser.

## Summary

Congratulations on creating your first app with Power Apps!

In this module, you discovered what Power Apps can do for your business and the building blocks of creating your first app. You then created an app from data in a Microsoft Excel workbook.

Additionally, you learned that:

- To create, share, and administer your apps, you will use make.powerapps.com, the Power Apps Studio, and the Power Apps Admin Center.
- The power of Power Apps comes from the ability to connect to related technologies in your business. Examples of these are Dataverse, Power Automate, Microsoft SharePoint, and other data sources.
- You can create an app by using several different methods. Some of these methods include from a template, a data source (like Microsoft SharePoint), or a blank canvas.

# Customize a canvas app in Power Apps

## Improve apps by making basic customizations in Power Apps

In the previous module, you generated the Flooring Estimates app and started to explore its default design. While the default screens make a useful app out of the box, you'll often want to customize a generated app to suit your needs.

This unit explains basic changes for each screen in the app. You can do a lot more to customize an app, but the best way to start learning is to take a generated app and make common customizations. This will allow you to become familiar with the controls, layouts, and functions.

Before you begin, watch the following video for a brief overview of what to expect when customizing your app.



<https://www.microsoft.com/videoplayer/embed/RE4vtmh>

### Browse screen

The Flooring Estimates app already shows an image and some text for each product, but the layout could be better.

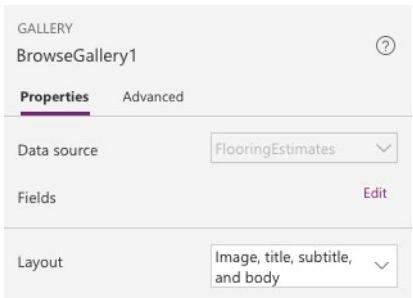
To improve the layout, use the following procedure:

1. On the Screens pane on the left, select **BrowseGallery1**.

The selection box around the gallery confirms your choice.

A screenshot of the Microsoft Power Apps studio interface. The top navigation bar shows 'FlooringEstimates' with icons for refresh, search, and add. Below is a search bar with placeholder 'Search items...'. The main area displays a 'BrowseGallery1' screen. It features a vertical list of six items, each representing a different carpet type with a small image, the name, and a brief description. The entire list is highlighted with a red rectangular selection box, indicating it is the active screen. The items are: Carpet Caserta Stone Beige, Carpet Caserta Sky Grey, Carpet Ageless Beauty Clay, Carpet Lush II Tundra, Carpet Lush II Frosty Glade, and Hardwood Bolivian Rosewood.

- 
2. On the right pane, open the Data pane by selecting the drop-down menu next to **Layout**.



3. Select a different layout, such as the one that shows the image, the title, and the subtitle but not the body.

GALLERY  
BrowseGallery1 (?)

**Properties** Advanced

Data source FlooringEstimates ▼

Fields Edit

Layout Image, title, and ▼

Search All ▼

List

Blank >

Title >

Title and subtitle >

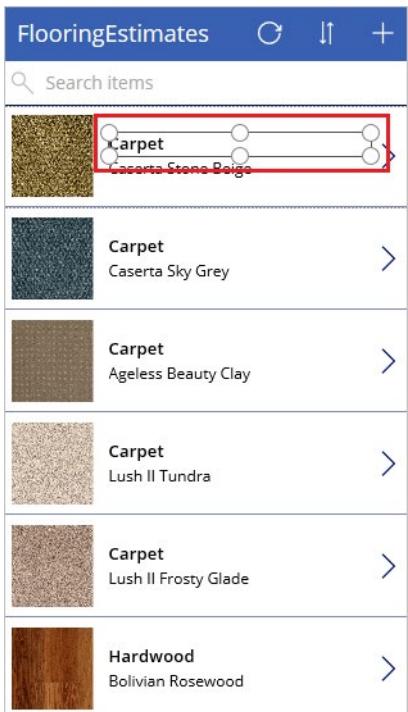
Title, subtitle, and body >

Image and title >

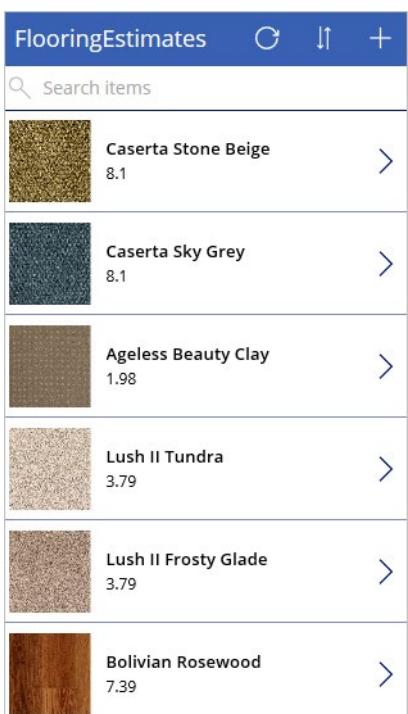
Image, title, and subtitle >

Image, title, and subtitle

4. Select the category of the item at the top of the gallery.



5. Change **ThisItem.Category** to **ThisItem.Name** in the formula bar.
6. Repeat the previous two steps but change the other **Label** control to show the price of each item.

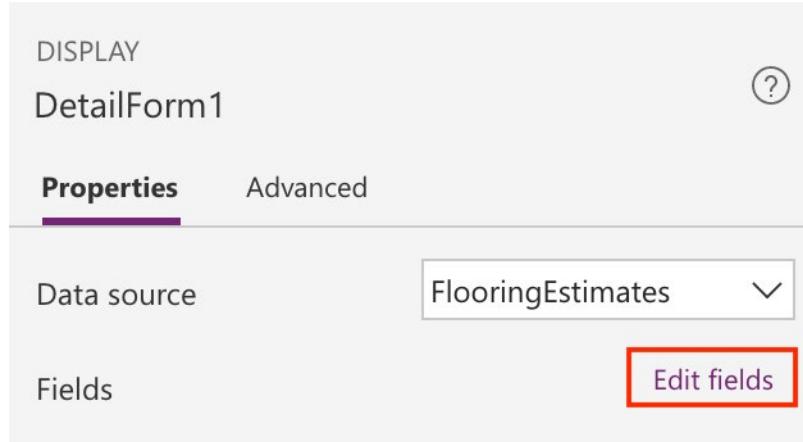


Changing the layout of a gallery and the types of data that it shows is that simple, and you might find that it's fun, too.

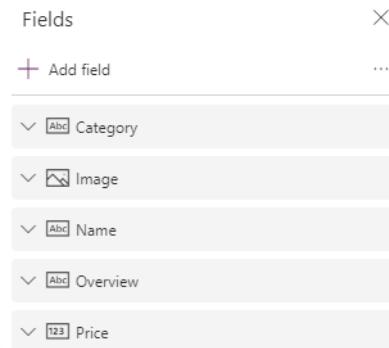
## Details screen

On the details screen, you want to change the order of the fields. The controls on this screen differ from the controls on the browse screen, so the process for changing them is also slightly different.

1. On the Screens pane on the left, select **DetailScreen1 > DetailForm1**.
2. On the right pane, select **Edit fields**.



3. Drag the **Name** field to the top of the list of fields and then drag the **Image** field to the bottom.



## Edit/create screen

On the screen where your users edit and create entries, you want to make it easier for them to enter information in a text box.

1. On the Screens pane on the left, select **EditScreen1 > EditForm1**.
2. On the right pane, select **Edit fields**.
3. Expand **Overview**. Select the drop-down arrow for the **Control type** and then select **Edit multi-line text**.

A multi-line edit control will simplify your user's ability to add more than a few words in this field.

A few basic steps can greatly improve the appearance and experience of using an app, and Power Apps Studio provides many options for customizing those apps.

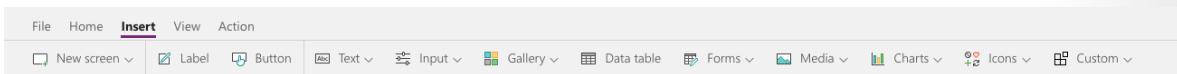
# Explore screens and controls in Power Apps

This unit examines the screens and other controls that define the behavior of apps that Microsoft Power Apps generates. All the details won't be covered; however, knowing more about how these apps work will help you build your own apps.

## Controls in Power Apps

A control is a UI element that produces an action or shows information. Many controls in Power Apps are similar to controls that you've used in other apps: labels, text-input boxes, drop-down lists, navigation elements, and so on.

In addition to these typical controls, Power Apps has more specialized controls, which you can find on the **Insert** tab.



A few controls that can add interest and impact to your apps include:

- **Galleries** - These controls are layout containers that hold a set of controls that show records from a data source.
- **Forms** - These controls show details about your data and let you create and edit records.
- **Media** - These controls let you add background images, include a camera button (so that users can take pictures from the app), a barcode reader for quickly capturing identification information, and more.
- **Charts** - These controls let you add charts so that users can perform instant analysis while they're on the road.

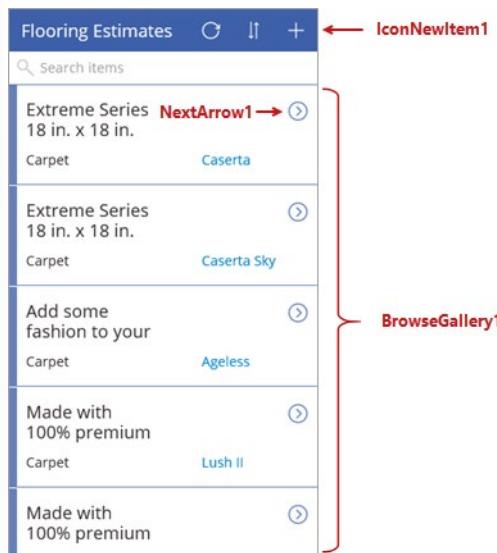
To see what controls are available, select the **Insert** tab, and then select each option in turn.

## Explore the browse screen

Each screen in the app has multiple controls, but one control takes up most of the screen space. The first screen in the app is the browse screen, which is named **BrowseScreen1** by default.

Controls in the browse screen that you'll want to become familiar with include:

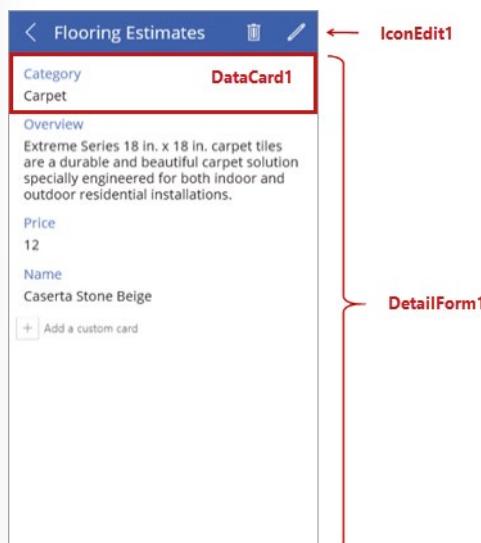
- **BrowseGallery1** - This control takes up most of the screen and shows data from your data source.
- **NextArrow1** - When this control is selected, it opens the details screen.
- **IconNewItem1** - When this control is selected, it opens the edit/create screen.



## Explore the details screen

The details screen is named **DetailScreen1** by default. Some of its controls are as follows:

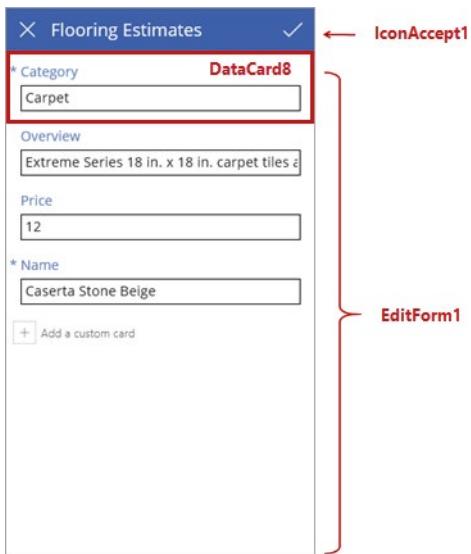
- **DetailForm1** - This control contains other controls and contains a data card for each field of the record that is being displayed.
- **DataCard1** - This is a card control. Each card represents a single field of the record. In this case, it shows a flooring category from the Flooring Estimates table, as shown in the previous unit.
- **IconEdit1** - When this control is selected, it opens the edit/create screen so that the user can edit the current item.



## Explore the edit/create screen

The third screen in the app is **EditScreen1**. Some of its controls include:

- **EditForm1** - This control contains other controls and contains a data card for each field of the record that is being edited.
- **DataCard8** - This is another card control that shows a flooring category from the Flooring Estimates table, as shown in the previous unit.
- **IconAccept1** - When this control is selected, it saves the user's changes.



## Excercise-Get started with functions in Power Apps

When using Microsoft Power Apps, you don't have to write complicated application code the way that a traditional developer does. However, you must express logic in an app and control its navigation, filtering, sorting, and other functionalities. This is where formulas come in.

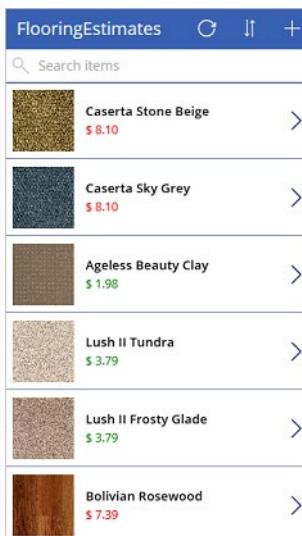
If you've used Microsoft Excel functions, you should recognize the approach that Power Apps takes. This unit shows a couple of basic formulas for text formatting and then describes three of the formulas that Power Apps includes when it generates an app. With this information, you'll have a better idea of what formulas can do, and then you can also start to write your own.

## Get started with formulas and properties

Properties determine the appearance and behavior of controls. Each type of control has a different set of properties.

The previous unit explored controls in all three screens of an app that Power Apps generated. In this section, you'll use the properties of the label control to format the price in the gallery.

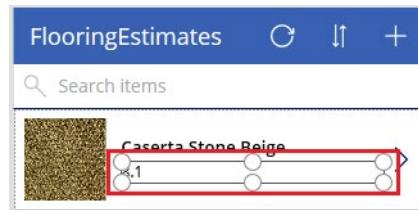
By default, the price appears as a plain number without a currency symbol. Suppose that you want to add a dollar sign and change the text color based on the item's cost (for example, red if it's more than \$5 but green otherwise). The following graphic shows the expected result.



By default, Power Apps pulls in a price value for each item. This value is set as the **Text** property of the label that shows the price.

**Note:** Be sure to complete the steps in unit 1 of this module as the description column is changed to price as reflected in the next steps.

1. In **BrowseScreen1**, select the price of the first item.



2. In the drop-down list of properties, select **Text**.
3. To add the currency symbol for US dollars, set the **Text** property to this formula:

Text(ThisItem.Price, "\$ ##.00")

```
> [<!NOTE>
> If your formula returns an error, then please note that the language
setting of your Power Apps environment can affect some separators and opera-
tors. For example, the above formula is expressed in a language and region
that uses dot or period as the decimal separator, such as Japan or the
United Kingdom. However, this same formula in a language and region where a
comma is used for the decimal separator, such as France or Spain, the
formula will need to be: `Text(ThisItem.Price; "$ ##,00")`
>
> The property selection operator . (dot or period) in ThisItem.Price is
always the same, no matter what the decimal separator is, but notice that
the decimal separator and the chaining operation separator changed to a
comma and semicolon respectively. Internally the formula doesn't change,
all that changes is how it's displayed and edited by the author. See <a
```

['Formula separators and chaining operator](/powerapps/maker/canvas-apps/global-apps?azure-portal=true#formula-separators-and-chaining-operator) for more information.

The \*\*Text\*\* function specifies how to format the number. The formula is like an Excel function, but Power Apps formulas refer to controls and other app elements instead of cells in a workbook.

If you select a control and then open the property drop-down list, a list of properties that are relevant to the control appears. For example, the following is a partial list of the properties for a \*\*Label\*\* control. Some properties are relevant across a wide range of controls, but others are relevant only for a specific control.



To conditionally format the price's color, set the \*\*Color\*\* property of the price's \*\*Label\*\* control to this formula:

`  
If(ThisItem.Price > 5, Color.Red, Color.Green)  
`

## Formulas included in the generated app

Power Apps uses a couple of formulas in every app that it generates. Both examples are from the browse screen and work with the \*\*OnSelect\*\* property. This property defines what happens when a user selects a control.

\* The first formula is associated with the \*\*IconNewItem1\*\* control.



Select this control to open the edit/create screen where you can create an item. To view the formula, select the  and then select it in the formula bar. The formula is as follows:

`NewForm(EditForm1);Navigate(EditScreen1, ScreenTransition.None)`

The formula instantiates an edit page on the edit/create screen so that users can create an item. A value of `ScreenTransition.None` means that there's no transition, such as a fade, between screens.

\* The second formula is associated with the \*\*IconSortUpDown1\*\* control . Select this control to sort the items in the gallery. The formula is as follows:

`UpdateContext({SortDescending1: !SortDescending1})`

The formula uses `UpdateContext` to update a variable called `SortDescending1`. The exclamation `!` symbol in the formula is a shortcut for the Not function. The value of the variable switches back and forth as you select the control. This variable tells the gallery on this screen how to sort the items.

The app contains many other formulas, so take some time to select controls and determine what formulas are set for various properties.

For more information about these and other functions, refer to [formula reference for Power Apps](https://docs.microsoft.com/powerapps/maker/canvas-apps/formula-reference) page.

For additional information on customizing a canvas app, refer to the Use the UI and controls in a canvas app in Power Apps learning path and the Use basic formulas to make a better canvas app in Power Apps learning path.

## Summary

This module showed you how to customize an app by adding controls, formatting, and logic in Power Apps.

Additionally, you were able to:

- Change the layout of a gallery.
- Change the data that a control shows.
- Change the order in which fields appear.
- Change the control with which a user provides information.
- Format a number as a price and coloring prices based on their values.

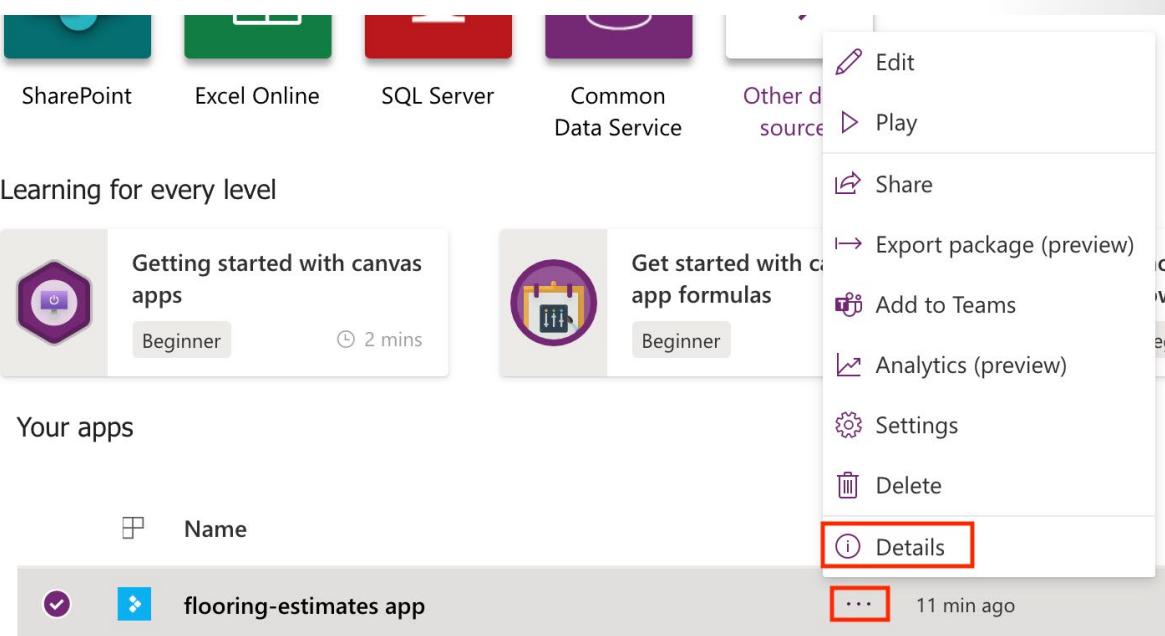
# Manage apps in Power Apps

## Exercise-Manage app versions in Power Apps

Microsoft Power Apps can help if you saved changes to an app that you shouldn't have or if something else goes wrong. For apps that you save in the cloud, Power Apps keeps a history of the changes that you make. You can view each version that you've saved and restore your app to a previous version if necessary. If you shared the app, the people whom you shared it with will also receive the restored version if you republish the app.

### View versions of your app

1. On [make.powerapps.com](https://make.powerapps.com)<sup>19</sup>, select **Apps** on the left pane.
2. In the list of apps, select the ellipsis (...) next to the app name and then select **Details**.

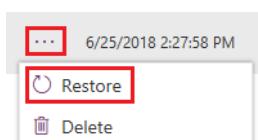


3. Select the **Versions** tab.

The tab shows all versions of the app that you saved as you developed it.

### Restore a previous version

1. Select the ellipsis (...) next to the desired version and then select **Restore**.



2. Select **Restore** again to confirm the action.

<sup>19</sup> <https://make.powerapps.com>

A new version is added to your list.

When you restore a version of an app, the newly restored version gets a new, incremented version number and appears at the top of the list. A new version never overwrites a previous version.

*Note:* After restoring a previous version, the restored version needs to be published before users will see the new version.

If for some reason you are not able to restore a previous version, you can try the following:

- Make sure the App is not open in Power Apps Studio. If the app is open, you will not be able to restore the version.
- Verify the version you would like to restore is not older than six months. At the current time, only app versions less than six months old can be restored.

## Exercise-Share apps in Power Apps

You can share an app with specific users, groups, or your whole organization. When you share an app with other people, they can run it in a browser, from the Microsoft Dynamics 365 home page, or in Microsoft Power Apps Mobile for Microsoft Windows, Apple iOS, or Google Android.

Even better, you can give someone permission to update the app.

### Prepare to share an app

To complete the following steps, open the app that you want to share in **Edit** mode.

1. In Power Apps Studio, select the **File** menu and then select **Settings**. Give the app a meaningful name and a description so that your team knows what your app does and can easily find it in the apps list.
2. On the **File** menu, select **Save as** and then select **The cloud**.  
You must save an app to the cloud before you can share it.
3. Select **Save** and then select **Share**.
4. On the **Share** tab, specify the users or groups with whom you want to share the app. To add everyone in your organization, type **Everyone** and select **Everyone in Company Name**. If you need to share with a large group of users, a best practice is to share through an Azure Active Directory Security Group.

By default, the user receives the User permission. If you want the user to also be able to edit the app, then

select the co-owner check box. The following is a description of both permissions:

- **Co-owner** - Users can use, edit, and share the app, but can't delete or change the owner.
- **User** - Users can view and use the app, but they can't change it.

Security-group considerations

- If you share an app with a security group, existing members of that group, and anyone who joins it, will have the permission that you specify for that group. Anyone who leaves the group loses that permission unless they belong to a different group that has access or if you give them permission as an individual.
- Every member of a security group has the same permission for an app as the overall group does. However, you can specify greater permissions for one or more members of that group to allow them greater access. For example, you can give Security Group A permission to run an app, but

you can also give User B, who belongs to that group, Co-owner permission. Every member of the security group can run the app, but only User B can edit it. If you give Security Group A Co-owner permission and User B permission to run the app, that user can still edit the app.

5. To notify users by email, leave the **Send an email invitation** check box selected.

If you elect to notify the users by email, everyone you shared the app with will receive an email message that has a link to the app. People whom you granted **Co-owner** permission for the app will also receive a link to Edit App in Power Apps Studio.

6. Select **Share**.

If you make and save changes to a shared app, the people whom you shared it with will see your changes as soon as you publish them. This can be useful if you improve the app, but it can also negatively affect users if you remove or significantly change features. Remember to create a notification plan for alerting your users of major updates.

## Permissions and licensing

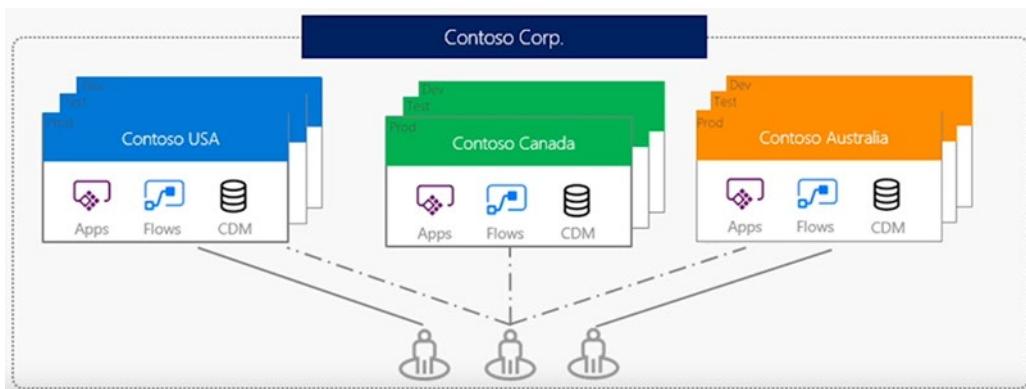
Basic information about permissions and licensing that you should be aware of are:

- Users and contributors need permissions to any data connections and gateways that a shared app uses. Some permissions come implicitly with the app, but you must explicitly grant others. If you create an app based on Dataverse, you must also ensure that the users with whom you share the app have the appropriate permissions for the table or tables on which the app relies. Specifically, those users must belong to a security role that can perform tasks such as creating, reading, writing, and deleting relevant rows. In many cases, you'll want to create one or more custom security roles with the exact permissions that users need to run the app. You can then assign a role to each user as appropriate.
- People who have **Co-owner** permission also need a Power Apps Per app plan or Power Apps Per user plan to work directly with tables in Dataverse.

Sharing an app is simple, and it's a great way to make an app that you find useful available to people across your organization.

## Exercise-Understand environments in Power Apps

An environment is a container for apps and other resources, such as data connections and flows from Power Automate. It's a way to group items based on business requirements.



If you've followed along with this module, you've already been working in [make.powerapps.com<sup>20</sup>](https://make.powerapps.com); therefore, you've been working in a specific environment the whole time. In the upper-right corner of the home page, you can view your current environment.



If you're new to Microsoft Power Apps, you might have only the default environment at this point. If a drop-down menu is visible next to the environment name, this indicates that other environments are available.

Note: If you want to work with Power Apps environments, you need a Power Apps Per app plan or Power Apps Per user plan. Additionally, if you want to work with Dynamics 365 restricted tables, you must have a Power Apps for Dynamics 365 license. Learn more about [licenses for Dynamics 365<sup>21</sup>](#).

## Reasons to use environments

Reasons to create environments beyond the default one include:

- **Separate app development by department** - In a large organization, each department can work in a different environment. That way, department employees see only apps and company data that are appropriate to their needs.
- **Support application lifecycle management (ALM)** - Separate environments let you separate apps that are in development stages from those that have already been shared. Alternatively, you might want to use a trial environment so that you can receive feedback from employees before publishing the final app. For some organizations, showing apps before they're completely developed and published can present security concerns.
- **Manage data access** - Each environment can have its own source of business data, called a database for Dataverse. Other data connections are specific to an environment and can't be shared across environments.

*Important:* Keep in mind that environments are relevant only to app creators and Power Apps admins. When you share an app with users, those users simply run the app, providing they have the correct permissions. In other words, they don't have to worry about what environment the app came from.

## Create an environment

Only an admin can create environments. If you aren't an admin, this information can still be helpful when you talk to your admin about setting up environments.

1. On the [make.powerapps.com<sup>22</sup>](https://make.powerapps.com) home page, select the gear icon near the upper-right corner and then select **Admin center**.

You can also go directly to <https://admin.powerplatform.microsoft.com/>

2. In the Power Apps admin center, select **+ New**.

<sup>20</sup> <https://make.powerapps.com>

<sup>21</sup> <https://na01.safelinks.protection.outlook.com/?url=https%3A%2F%2Fdocs.microsoft.com%2Fen-us%2Fpowerapps%2Fadministrator%2Fpricing-billing-skus%23licenses&data=02%7C01%7Cv-tosis%40microsoft.com%7C3bb58d639d8745c27ff908d62d4c1062%7C72f988bf86f141af91ab2d7cd011db47%7C1%7C0%7C636746202048937165&sdata=8rDKLL4XBkwCLOrpZe0F2MITmkfhAmukrV3bn4a34NU%3D&reserved=0>

<sup>22</sup> <https://make.powerapps.com>

3. In the **New environment** dialog box, enter a name for the environment and then select a region and an environment type.
4. To the left of **Create a database for this environment**, select the toggle to Yes.
5. Select **Next**.
6. Select the currency and language for the data that is stored in the database. You cannot change the currency or language after the database is created.
7. Select **Save**.

It might take several minutes to create the database on Dataverse. After the database is created, the new environment appears in the list of environments on the **Environments** page.

You now have a new environment to work in. If you go back to [make.powerapps.com<sup>23</sup>](https://make.powerapps.com), you will see it in the environments list.

## Manage access to an environment

By default, you can access an environment in one of two ways:

- **System admin** - A system admin has full permissions to create and manage environments.
- **Environment maker** - An environment maker can view all apps in that environment, create apps, and work with Dataverse (other permissions apply).

Environment admins can create other security roles as needed. They can also add and assign users to these roles.

1. Start by going to [make.powerapps.com<sup>24</sup>](https://make.powerapps.com)
2. On the left pane, **Environments** should be selected by default, if it is not, select **Environments**.
3. Select the test environment that you just created, and then select **Settings** at the top.
4. Select the **Users + permissions** dropdown and select **Users**.
5. Select **Add user** at the top and add the user by entering the email address of the user in your organization and then selecting **Add**. Wait a few minutes for the user to be added.
6. To manage the roles and information of a user, select the user's **Name**. This will open a new tab with the Dynamics 365 view of that user.
7. Select **Manage Roles** on the top bar.
8. In the **Manage User Roles** box, select the role(s) for the user. In this example, assign the user to the Environment Maker role.
9. Select **OK**.
10. The changes are then saved, so you can close the Dynamics 365 tab in your browser when done.

## Power Apps review

Congratulations on building your first app!

<sup>23</sup> <https://make.powerapps.com>

<sup>24</sup> <https://make.powerapps.com>

To review, so far you've learned how to:

- Build an app based on data in a Microsoft Excel workbook that's stored in Microsoft OneDrive for Business. You learned that Microsoft Power Apps can connect almost as easily to data sources that you're already using in the cloud (such as Microsoft SharePoint, Microsoft Azure, Google Drive, and Salesforce) or on-premises.
- Customize an app to make it your own by modifying the appearance and behavior and adding Excel-like functions.
- Share apps instantly with your co-workers across the web, tablets, and mobile devices.
- Set up environments so that you can separate a working environment from the one that you want to share with your team.

The best way to advance your skills is to run the samples, practice using the templates, and generate more apps from your own data.

## Next steps

One goal of this module is to provide a clearer picture of what Power Apps is and how you can start creating apps, regardless of your experience level. The following are useful resources and downloads to help enhance your further learning.

## Power Apps resources

- Explore further with the **Power Apps documentation<sup>25</sup>**.
- Stay current with the **Power Apps blog<sup>26</sup>**.
- Join the **Power Apps community<sup>27</sup>**.
- Expand your expertise with additional Power Apps learning paths:
  - **Use basic formulas to make better Power Apps canvas apps<sup>28</sup>**
  - **Work with data in a Power Apps canvas app<sup>29</sup>**
  - **Use the UI and controls in a canvas app in Power Apps<sup>30</sup>**
  - **Use advanced data options and connectors in Power Apps<sup>31</sup>**
  - **Master advanced techniques for Power Apps canvas apps<sup>32</sup>**
- Improve Power Apps by submitting an **idea<sup>33</sup>**.

## Power Apps downloads

- **Power Apps Mobile for Windows<sup>34</sup>**

---

<sup>25</sup> <https://docs.microsoft.com/powerapps/>

<sup>26</sup> <https://powerapps.microsoft.com/blog/>

<sup>27</sup> <https://powerusers.microsoft.com/t5/PowerApps-Community/ct-p/PowerApps1>

<sup>28</sup> <https://docs.microsoft.com/learn/patterns/use-basic-formulas-powerapps-canvas-app/>

<sup>29</sup> <https://docs.microsoft.com/learn/patterns/work-with-data-in-a-canvas-app/>

<sup>30</sup> <https://docs.microsoft.com/learn/patterns/ui-controls-canvas-app-powerapps/>

<sup>31</sup> <https://docs.microsoft.com/learn/patterns/advanced-data-options-and-connectors/>

<sup>32</sup> <https://docs.microsoft.com/learn/patterns/understand-advanced-topics/>

<sup>33</sup> <https://powerusers.microsoft.com/t5/PowerApps-Ideas/idb-p/PowerAppsIdeas>

<sup>34</sup> <https://aka.ms/powerappswin>

- 
- **Power Apps Mobile for iOS<sup>35</sup>**
  - **Power Apps Mobile for Android<sup>36</sup>**

## Summary

In this module, you learned about app versions, sharing apps, and managing environments.

Additionally, you learned how to:

- View and restore app versions
- Share an app and manage its permissions
- Create environments and how to manage security access

---

<sup>35</sup> <https://aka.ms/powerappsios>

<sup>36</sup> <https://aka.ms/powerappsandroid>

# Navigation in a canvas app in Power Apps

## Understand navigation

In Power Apps, because most apps have multiple screens, it's essential that you understand how to implement the **Navigate** function in your app. The **Navigate** function allows users to move or navigate from screen to screen in an app. For example, if you create an app with three screens and you want to give users a way to navigate to another screen, set the **OnSelect** property of a **Button** control to:

**Navigate(Screen2,ScreenTransition.Cover)**

With this formula, when the button is selected, **Screen2** will automatically display. You could also use this formula on an icon, like an arrow, to provide navigation in the app.

The **Navigate** function also allows for a visual transition as the users move from screen to screen, which is set using **ScreenTransition**. In the example shown above, **ScreenTransition.Cover** was applied. There are a few different **ScreenTransitions** to choose from, and each one provides a slightly different user experience when navigating screens. **ScreenTransitions** will be covered in further detail later in this module. Similar to the **Navigate** function, you also have the option to use the **Back()** function. The **Back()** function is fairly straight-forward, it sends the user back to the previous screen.

You can use **Navigate** to set one or more context variables. You can use this approach to pass parameters to a screen. If you've used another programming tool, this is similar to passing parameters to procedures.

## Hidden screens

You can have "Hidden screens" in your app for various purposes, such as:

- Documentation screen
- Settings screen
- Special permissions screen

A "Hidden screen" allows the creator of the app to add screens but not give users to access those screens. This is accomplished by not creating any navigation for users to those screens. There's an example later in this module to demonstrate this functionality further.

## The navigation and back function

In the previous section, you learned how to implement the **Navigate** and **Back()** functions using a Control. When implementing these functions, like any other function in Power Apps, there are certain arguments that need to be defined.

## Navigate function

Here's a breakdown of the syntax for the navigation function:

**Navigate(Screen, Transition [, UpdateContextRecord ] )**

- **Screen** - Required. The name of the screen to display.
- **ScreenTransition** - Required. The visual transition to use between the current screen and the next screen.
- **UpdateContextRecord** - Optional. A record that contains the name of at least one column and a value for each column. This record updates the context variables of the new screen. For more information, see [UpdateContext function in Power Apps<sup>37</sup>](#).

In the first argument, you specify the name of the screen to display. In the second argument, you specify how the old screen changes to the new screen. In the third argument, you have the option to specify a Context variable. Looking back to the second argument, ScreenTransition, there are a number of different ScreenTransitions you could apply. Each ScreenTransition produces a slightly different visual experience for the user.

## Screen transitions

**ScreenTransition.Cover** - The new screen slides into view, covering the current screen.

**ScreenTransition.Fade** - The old screen fades away to reveal the new screen.

**ScreenTransition.None** - The old screen is quickly replaced with the new screen.

**ScreenTransition.UnCover** - The old screen slides out of view, uncovering the new screen.

Here are some examples using ScreenTransitions.

| Formula                                                   | Description                                                                                                                              | Result                                                                                                                                  |
|-----------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| <b>Navigate(Details, ScreenTransition.None)</b>           | Displays the <b>Details</b> screen with no transition or change in value for a context variable.                                         | The <b>Details</b> screen appears quickly.                                                                                              |
| <b>Navigate(Details, ScreenTransition.Fade)</b>           | Displays the <b>Details</b> screen with a <b>Fade</b> transition. No value of a context variable is changed.                             | The current screen fades away to show the <b>Details</b> screen.                                                                        |
| <b>Navigate(Details, ScreenTransition.Fade, {ID: 12})</b> | Displays the <b>Details</b> screen with a <b>Fade</b> transition, and updates the value of the <b>ID</b> context variable to <b>12</b> . | The current screen fades away to show the <b>Details</b> screen, and the context variable <b>ID</b> on the screen is set to <b>12</b> . |

The **Back ()** function has an optional argument for ScreenTransition.

<sup>37</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/functions/function-updatecontext>

Here's a more detailed example using these functions with multiple controls in a common real-world scenario. In many of the apps that you develop, certain screens may have multiple controls allowing users to navigate to different screens depending on the control they select. In the following example, you will create a three screen app to demonstrate the **Navigate** and **Back()** functionality.

1. In Power Apps Studio, create 3 blank screens.
2. On Screen1, add a **Button** control and change the **Text** property to **Next**.
3. On Screen2, add two **Button** controls and change the **Text** of one button to **Next** and the other button control to **Back**.
4. On Screen3, add a **Button** control and change the **Text** property to **Back**.
5. On Screen1, set the Next button **OnSelect** property to **Navigate(Screen2,ScreenTransition.Fade)**.
6. On Screen2, set the **OnSelect** property for the Next button to **Navigate(Screen3,ScreenTransition.Cover)**.
7. Set the **OnSelect** property for the Back button to **Back()**.
8. On Screen3, set the **OnSelect** property for the Back button to **Back()**.
9. To test this, put the app in Preview or Play mode and navigate through the app as a user would.

As you select each button, notice the subtle visual transitions of each ScreenTransition. Remember, the Navigate function must include a ScreenTransition. If you tried to write your Navigate function like this, **Navigate(Screen2)**, Power Apps would display a red squiggly line in the formula box indicating that there is an issue with the function as written.

## Back () function

When using the **Back()** function it should be noted, this will take you back to the previous screen that you navigated from, the key word being "navigated". To elaborate, this does not mean the screen you or the user was just on, you need to use the **Navigate** function to navigate away from the screen to then use the **Back()** function to get back there. This can be a little confusing, so here's an example of how this works.

In the previous example, there are three screens. Update the **OnSelect** property of the Next button on Screen1 from **Navigate(Screen2,ScreenTransition.Cover)** to **Navigate(Screen3,ScreenTransition.Cover)** and the **text** property to **Jump to Screen 3**. Now, when the user selects the button, the app will navigate to Screen3. Then when they select the Back button on Screen3, they will navigate back to Screen1 and skip Screen2 entirely. This is by design in Power Apps and is an important concept to ensure the navigation in your app is configured properly.

Before moving on to the next section, add one more screen to the example app and rename the screen **Documentation**. With no Navigation pointing to this screen, it is not accessi-

ble to your end users. The purpose of this screen is to give the App creator a location in the app to make notes or add documentation about certain aspects of how the app functions. The App creator can also provide instructions for other editors of the app so they can quickly and easily identify what the previous creator did.

In this simple four screen app example, you can see how easy it is to navigate screens by setting the **OnSelect** property of a control.

There are several additional ways to navigate through your app covered in the next section.

## More ways to use navigation function

As previously mentioned, one of the most common ways to move from screen to screen is by setting the OnSelect property of a control. Keep in mind that there are also several other ways to trigger the Navigate or Back() functions.

### Navigating screens by setting the **OnChange** property of a drop-down control

One option is to use a **DropDown** control and an If statement.

With this solution, depending on the choice selected in the drop-down menu the app navigates to a specific screen.

1. In Power Apps Studio, add 2 blank screens. There should be a total of 3 blank screens.
2. On Screen1, add a **dropdown** control.
3. Set the **Items** property for the **dropdown** control to

```
[" ", "Active", "Inactive"]
```

4. Set the **OnChange** property for the **dropdown** control to the following

```
If (Dropdown1.Selected.Value = "Active", Navigate(Screen2, ScreenTransition.Cover),
If (Dropdown1.Selected.Value = "Inactive", Navigate(Screen3, ScreenTransition.Fade)))
```

On your keyboard, hold the Alt key to test the new functionality.

When the user selects **Active** from the drop-down menu, the user will be sent to Screen2. When the user selects **Inactive** from the drop-down menu, the user will be sent to Screen3. Finally, if the blank option is selected from the drop-down menu, the app will not navigate to another screen.

### Using variables and the Timer control to navigate screens

You can use a variable and an If statement to set navigation. Depending on what the variable is set to this will send the user to a specific screen. For example, if you have the question "Do you have additional feedback to provide regarding this incident?" If the user selects "Yes", then you want to open the **Additional Information** screen. If the user selects, "No", then you want to load the **Form Completed** screen.

Another option is to use a timer control. When the time runs out, the app navigates to a different screen. For example, maybe you only want to allow a user 30 seconds to answer the questions on a screen before navigating to the next set of questions.

The following example builds off the previous example and incorporates variables and a Timer control.

1. On Screen1, add a **Timer** control.
2. Set the **Duration** property to 10000 (milliseconds).
3. Set the **Auto start** to **On**.
4. Set the **OnTimerEnd** property to

```
If(Dropdown1.Selected.Value = " ", Navigate(Screen2, ScreenTransition.None))
```

5. Select the drop-down control and change the **OnChange** property to

```
If(Dropdown1.Selected.Value =
"Active", Set(varStatus,1), If(Dropdown1.Selected.Value =
"Inactive", Set(varStatus,2), Set(varStatus,0)))
```

6. Add a **Button** control under the drop-down menu, and set the **Text** property to **Next**.

7. Set the **OnSelect** property for the button to

```
If(varStatus = 1, Navigate(Screen2, ScreenTransition.Cover),
If(varStatus = 2, Navigate(Screen3, ScreenTransition.Fade)))
```

Now, if the Timer Ends, the user is automatically sent to Screen two. If the user selects **Active** from the drop-down menu, a Variable named varStatus is set to 1. When the user selects the **Next** button, they will be sent to Screen2. If the user selects **Inactive** from the drop-down menu, a Variable named varStatus is set to 2. When the user selects the **Next** button, they will be sent to Screen3.

Test this by putting the app in Preview or Play mode, select the Timer control and wait for 10 seconds.

Now test the variables by selecting **Active** or **Inactive** from the drop-down menu and selecting the **Next** button.

As you can see there are different ways to configure navigation in your app. In the previous example, the navigation automatically changed when the user selects a value in the drop-down control. In the second example, navigation can happen in two different ways: when the timer reaches zero or when the user selects a button. The variable still allows you to control which screen they navigate to based on the drop-down value selected, but they will not automatically navigate there when the value is selected. You have the flexibility to choose what works best in your solution.

## Summary

In Power Apps, you can use a number of different controls, variables, and other functions to fully customize the end user's navigation experience.

- Use the **Navigate** function to send the user to any screen in your app.

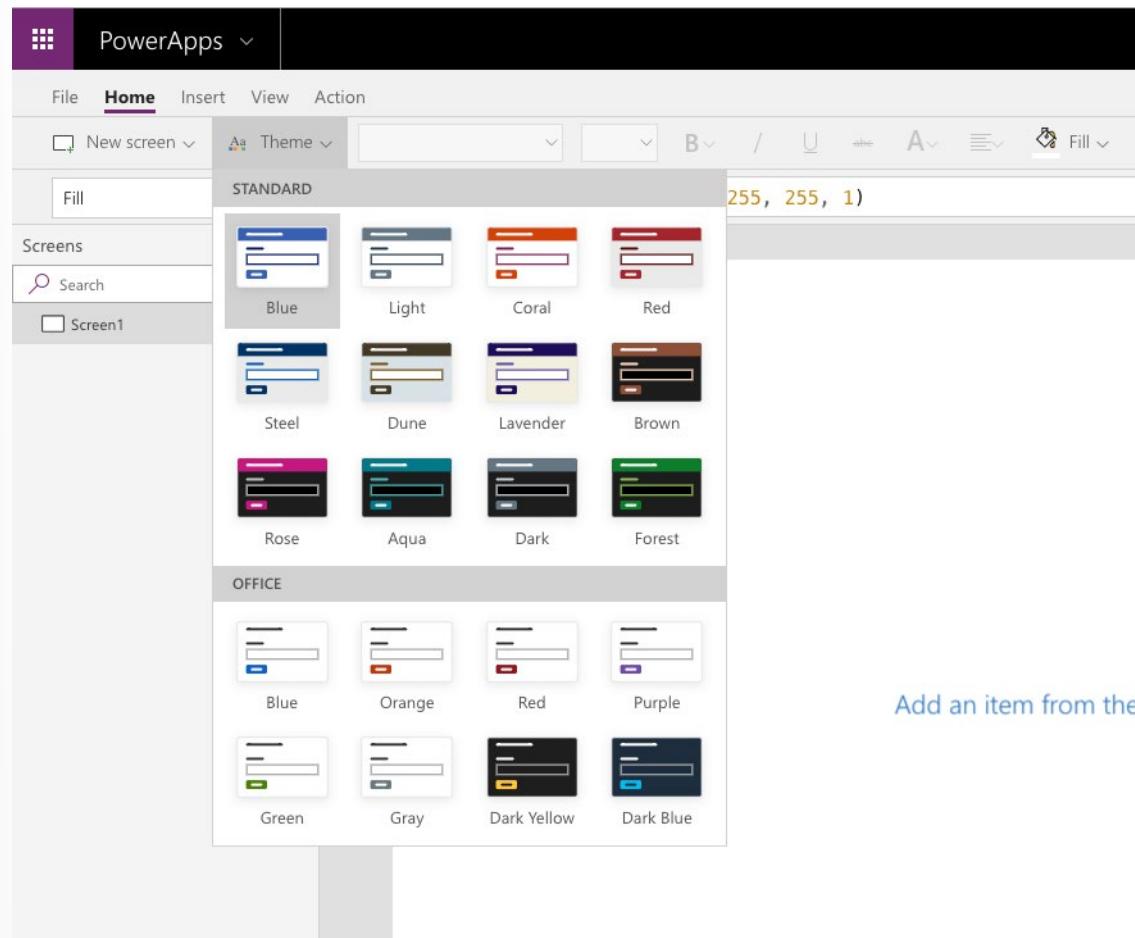
- Create additional navigational functionality by utilizing If statements, variables, and timer controls.
- Use the Back() function to send your end-users back to the screen that they previously navigated from.

By choosing to use a combination of different controls, properties, functions, and formulas you can develop and deploy an app that's both functionally sound and simple to use. It should also be noted, just because you can add navigation in many ways doesn't mean it's always the best to do so. Think about the end-user interacting with the app and keep it simple.

# How to build the UI in a canvas app in Power Apps

## Use themes to quickly change the appearance of your app

A quick and easy way to change the colors in your app is to apply a theme. In Power Apps, there are several out-of-the-box themes to choose from. The following screenshot illustrates all the themes that are available in your app.



These themes have a specific set of default colors and visual elements which will alter the look and feel of your entire app. To use any of these themes in your Canvas app, select a theme from the ribbon. The changes to your app will take effect immediately.

If none of the included themes work for your app, you can create a custom theme. For example, if you decide to select the Steel theme but want the screen background to be a lighter color, this can be easily done. To change the background of your screen, change the **Fill** property to **RGBA(250, 250, 250, 1)**. The

screen will be a slighter lighter shade of gray. Keep in mind, this only changed the fill of that specific screen, if you want to add a new blank screen it would still have the **Fill** property of **RGBA(232, 232, 232, 1)**. This is by design when working with a canvas app. Also, there is

no option to create a Custom theme for a canvas app and store it to be used for other apps.

Typically, many organizations will select the theme that best suits their needs, and then alter the various control properties to align with their corporate branding.

## Branding a control

As noted earlier, one of the built-in themes may not match your organization's desired look and feel for the app. You can customize your app by changing various properties of the app controls. By adjusting a few of the Control properties, like Fill, Hover, and Border you can completely change how the control looks. If you decide to customize your controls, it's recommended that you do this thorough testing to ensure that you don't run into any complications when users interact with the app.

For example, consider the Button control. The following are some of the properties of a Button control that you could customize to better fit your company's theme.

## Typical properties

These properties are in effect when the user is not interacting with the control.

- BorderColor - The color of a control's border.
- BorderStyle - Determines whether a control's border is solid, dashed, dotted, or none.
- Color - The color of text in a control.
- Fill - The background color of a control

## Disabled properties

These properties are in effect when the control is disabled. A control can be disabled if the **Disabled** property is set to **Disabled**.

- DisabledColor - The color of text in a control if its **DisplayMode** property is set to **Disabled**.
- DisabledFill - The background color of a control if its **DisplayMode** property is set to **Disabled**.

## Hover properties

These properties are in effect when the user hovers over the control with a mouse.

- HoverColor - The color of the text in a control when the user keeps the mouse pointer on it.
- HoverFill - The background color of a control when the user keeps the mouse pointer on it.

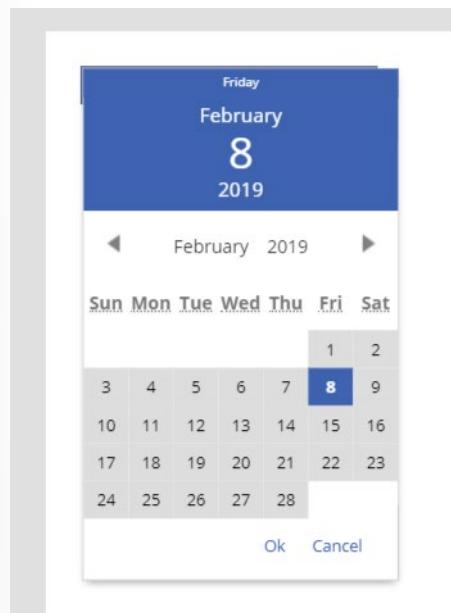
These are just some of the properties that you could modify. For more details about the properties that you can customize, see **Color and Border Properties in Power Apps**<sup>38</sup>.

Note that each control is independent. This means that if you alter the **HoverColor** property of one button control on your screen or in your app, the other buttons in the app will remain unchanged. You must edit the properties of each control that you want to appear in a different manner.

## Some color settings are only controlled by the theme

There are certain aspects of controls that cannot be altered and are specific to the theme that you select. For example, here's an example of the **Date picker** control.

1. In Power Apps Studio, add the **Date picker** control.
2. Put the app in preview mode and select the control so that it opens.
3. The color at the top of the **Date picker** control is specific to the theme, meaning there isn't a control property you can change to set the color manually. To change the color of that background you would have to select a different theme.



As you design your app, be sure to incorporate the use of icons. In the next section, you'll learn how to add an icon to a Canvas app to change the look and feel of the app.

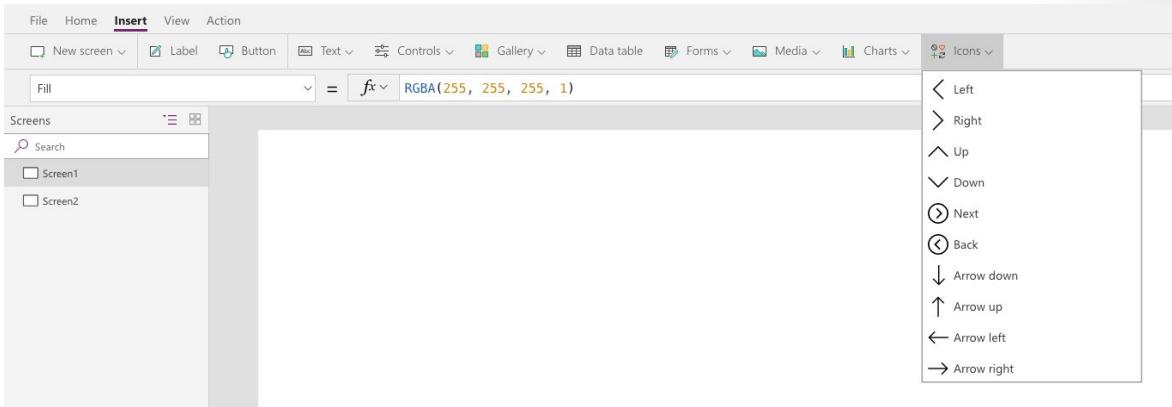
## Icons

When designing your app, utilize Shapes and Icon controls when possible to enhance the user experience of your app. There are certain shapes and icons that are universally recognized that you will find in many of the apps that you use daily. For example, instead of adding a Button

<sup>38</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/controls/properties-color-border>

control to your app and setting the **OnSelect** property to Back(), you could as easily use the **Back** icon control and set its **OnSelect** property to Back().

Remember, icons are controls, and each control has a specific set of properties that can be modified to change how the control looks and functions. To view all of available icon controls, select **Insert** and then select the **Icons** drop-down menu.



These controls include arrows, geometric shapes, actions, and symbols. By incorporating shapes and icons, you can save some space and reduce clutter in your app, especially when working with a Phone form factor app. The Phone form factor app has a much smaller area for designing and adding controls so replacing some of the buttons with shapes and icons could really benefit the overall spacing of your app.

## Images

The Image control is a control that shows an image. The image may be from a local file or a data source. Adding an image, such as a company logo is an easy way to enhance the overall look and feel of your app. On each screen you can apply a Background image, as well as multiple Image controls. You're not limited to a certain number of images that you can display; you can have as many images as you would like. Too many images may cause issues with the app performance and load times, but you would have to add quite a few large images for this to be an issue.

There are a set of properties specific to the Image control that allow you to customize how the image is displayed. When working with the Image control, a few of the more common properties you will most likely want to modify are the Image, Image position, and Transparency properties.

**Image** - The name of the image that appears in an image, audio, or microphone control.

**Image Position** - The position (Fill, Fit, Stretch, Tile, or Center) of the image in a screen or a control if it isn't the same size as the image.

**Transparency** - The degree to which the controls behind an image remain visible.

Here's a quick example of how to change the transparency and image position of an Image control.

1. In Power Apps, create a Tablet app using the "Product Showcase" App template.
2. On the left, under **Screens**, select **Image1**.

3. In the right pane, set the **Transparency** property to **0.5**.

You will notice that the image immediately becomes lighter in color.

4. In the **Image position** property, change this to **Center**.

The image appears to be zoomed in or larger.

As you develop your app, keep these settings in mind when using the Image control to enhance the look and feel of your app.

## Personalization

In Power Apps, you can show information about the current user with the **User()** function. This includes the full name, email address, and the picture that's associated with the user who's signed into a canvas app. It will match the "Account" information that is displayed in the Power Apps players and studio, which can be found outside of any authored apps. This may not match the current user's information in Office 365 or other services.

The **User** function returns

a **record<sup>39</sup>** of

information about the current user:

| Property               | Description                                                                                  |
|------------------------|----------------------------------------------------------------------------------------------|
| <b>User().Email</b>    | Email address of the current user.                                                           |
| <b>User().FullName</b> | Full name of the current user, including first and last name.                                |
| <b>User().Image</b>    | Image of the current user. This will be an image URL of the form "blob: <i>identifier</i> ". |

Set the **Image property<sup>40</sup>** of the **Image control<sup>41</sup>** to this value to display the image in the app.

Here's an example of how to add a user's profile picture, email, and name to your app.

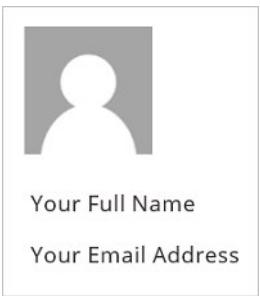
1. On the **Insert** tab, click or tap **Media**, and then click or tap **Image**.
2. Set the **Image** property to this formula: **User().Image**
3. On the **Insert** tab, click or tap **Text**, and then click or tap **Label**:
4. Set the **Text** property to this formula: **User().FullName**
5. Move the label so it's below the image control.
6. Add another label, and set its **Text** property to this formula: **User().Email**
7. Move the label so it's below the first label:

---

<sup>39</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/working-with-tables#records>

<sup>40</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/controls/properties-visual>

<sup>41</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/controls/control-image>



## Using the tablet or phone form factors

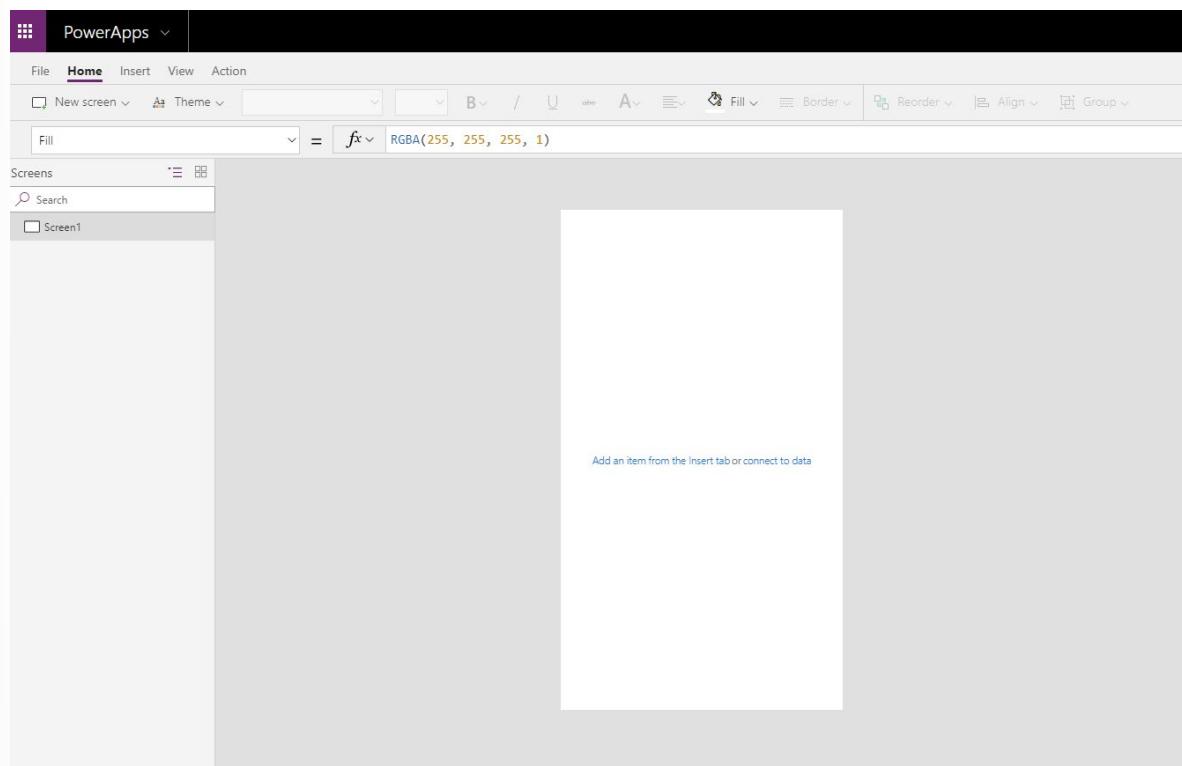
It is important to design your app for the primary device that it will be used on. There are two form factors to choose from, the Phone form factor and the tablet form factor. The main difference between these is the Screen size. The screen size affects the amount of space available to build the app.

The Phone form factor has a significantly smaller area to build your app, but if most of your users will be accessing the app from a mobile phone then this is the best form factor for you. When building for mobile, select controls that will be easy to use on a mobile device, ensure that the text is large enough to be easily seen, and design the app in a single column vertical format.

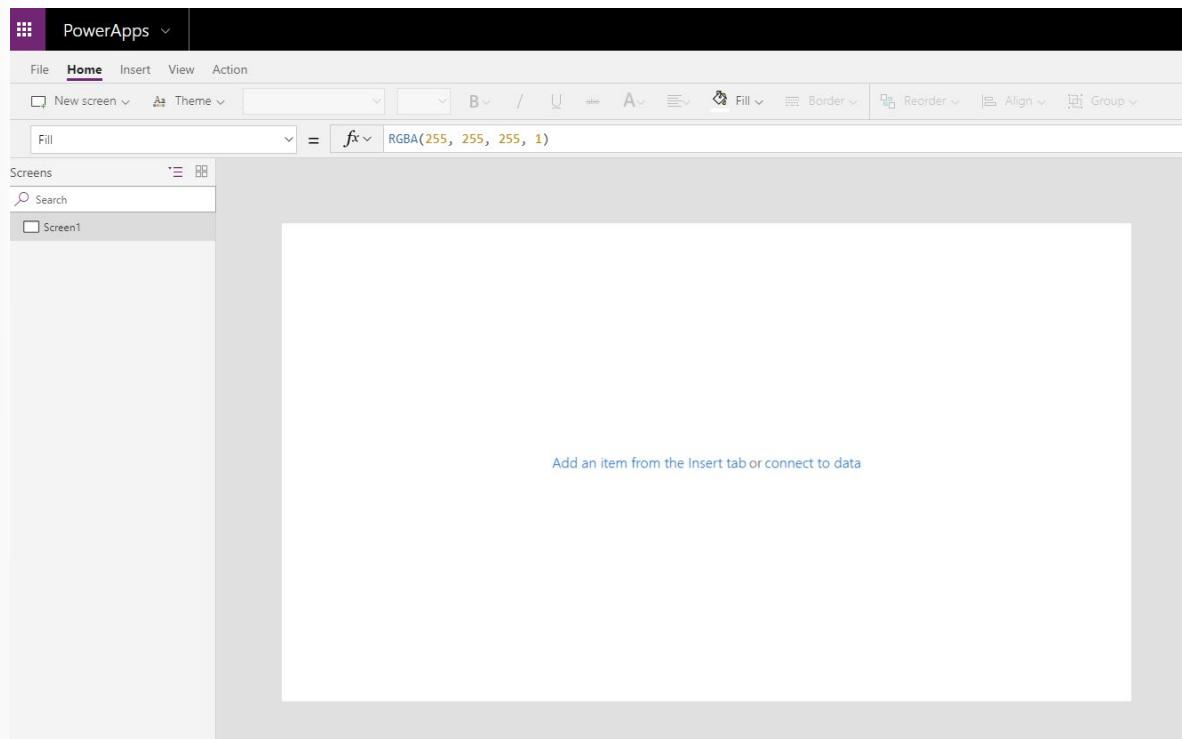
The tablet form factor has a much larger area for designing your app and is the best option if your users will be accessing the app from a tablet or PC. Because you have more screen size to work with, you will have more flexibility in designing this app. Regardless of the form factor that you choose; the functionality available in the Power Apps Studio will be the same.

Take a moment and create two blank Canvas apps. For one of the apps use the Phone form factor and for the other app use the tablet form factor. You will immediately notice the difference in the amount of space available on the screen to design your app.

### **Phone form factor**



### tablet form factor



Depending on the form factor that you choose, you have the ability to alter the screen size. To view the current screen size and orientation, for either form factor, see following steps.

1. In Power Apps Studio, in the upper-left corner, select **File**.

---

2. On the left, select **App settings**.

When you are in the tablet app, you will notice multiple options to select from for the screen size. Be sure to select the appropriate size for the majority of your app users.

- 16:9 (Default)
- 3:2 (Surface Pro 3)
- 16:10 (Widescreen)
- 4:3 (iPad)
- Custom

The Phone form factor does not include screen size options.

## Lock aspect ratio and lock orientation

Other options that you might want to consider regarding the screen size and orientation include the lock aspect ratio and lock orientation settings.

If you lock the aspect ratio, the app will retain the appropriate aspect ratio for a phone. If the app is running on another kind of device, the app will display incorrectly and may show unwanted results. If you unlock the aspect ratio, the app will adjust to the aspect ratio of the device on which it's running.

If you lock the app's orientation, the app will retain the orientation that you specify. If the app is running on a device for which the screen is in a different orientation, the app will display incorrectly and may show unwanted results. If you unlock the app's orientation, it will adjust to the screen orientation of the device on which it's running.

You can also modify the app's orientation by enabling **Enable app embedding user experience** in **Advanced settings**. This feature aligns the app when it's embedded and changes the background color of the hosting canvas to white.

## Summary

There are many options available to enhance the usability and appearance of your app. Throughout this module, you explored themes, properties for changing a control's appearance, using icons and images, personalization, and the different form factors available. Some important items to remember are:

- There are several out-of-the box or default themes to choose from and these can be easily applied.
- By modifying various Control properties, you can completely change how the control looks and functions when users interact with the app.
- Use the User() function to personalize the app. For example, create a Welcome screen and greet the user by using the User() function.

- You must select the form factor when you begin building your app. After you have selected tablet or mobile, it cannot be changed. With the tablet form factor, you can adjust the screen size.

# Use and understand Controls in a canvas app in Power Apps

## Understand controls

In Power Apps, you can add a variety of user interface (UI) elements to your canvas app and configure aspects of their appearance and behavior directly from the toolbar, Properties tab, or formula bar. These UI elements are called **Controls**. Many of the controls in Power Apps are just like the controls that you've used in other apps, such as labels, text-input boxes, drop-down lists, and navigation elements.

You can find all the controls available in Power Apps on the **Insert** tab.



You can configure the appearance and behavior of a control by setting one or more of its properties. Each type of control has a different set of properties. Some properties, such as **Height** and **Width**, are common to almost every type of control, but other properties, such as **ChevronFill** are specific to only certain Controls.

In addition to the controls listed above, there are other types of controls that you can add to enhance your apps:

**Galleries** - These controls are layout containers that hold a set of controls that show data from a data source. For more information about galleries, see [Work with data in a Power Apps canvas app<sup>42</sup>](#).

**Data table** - The **Data table** control shows data from a data source in a format that includes column headers for each column that the control shows. As an app maker, you have full control over which columns appear and in what order. Like the **Gallery** control, the **Data table** control maintains a **Selected** property that points to the selected row. Therefore, you can link the **Data table** control to other controls.

**Forms** - These controls show details about your data and let you create and edit items. For more information about forms, see [Write data in a Power Apps canvas app<sup>43</sup>](#).

**Media** - These controls let you add background images and sounds. Controls include a camera button (so that users can take pictures from the app) and a barcode reader for quickly capturing identification information.

**Charts** - These controls let you add charts so that users can do instant analysis.

**Icons** - These controls include shapes, graphics, and symbols to enhance the user interface. They are quickly recognizable by your users to ease their interaction with the app. For more information about icon controls, see the [How to build the UI in a canvas app in Power Apps](#) module of this learning path.

<sup>42</sup> <https://docs.microsoft.com/learn/patterns/work-with-data-in-a-canvas-app/>

<sup>43</sup> <https://docs.microsoft.com/learn/modules/write-data/>

In this module, you will learn about some of these controls and their properties to see how you can incorporate them into your app to enhance the overall functionality. Before developing your app, take a moment to determine the functionality that you want to provide and then select the control that best fits those needs. The more familiar you are with controls and how to work with them, the easier it will be to design your app.

In some cases, certain controls are almost interchangeable, and in those cases, you can use the control that you prefer. For example, the **Dropdown** and **Combo box** are similar controls. One key difference between them is that a **Combo Box** allows you to search for items as well as select multiple items. The **Dropdown** control does not support this functionality.

## Core properties of controls

Controls are one of the biggest components of a Power App because implementation determines how your app interacts with your data, what actions are available to a user, and what conditions may need to be met before another action can be taken. You can configure the appearance and behavior of a control by setting one of its properties. Because controls are designed with specific use cases in mind, the properties for each control are slightly different. Here are some important properties to be aware of:

- **Default** - The initial value of a control before it is changed by the user. For example, when working with a Drop down control you could set the default value to appear when users see the control.
- **DelayOutput** - Set to true to delay action during text input.
- **DisplayMode** - Values can be **Edit**, **View**, or **Disabled**. Configures whether the control allows user input (**Edit**), only displays data (**View**), or is disabled (**Disabled**). For more information about display modes, see Use basic formulas to make better canvas apps in Power Apps learning path.
- **Items** - The source of data that appears in a control such as a gallery, list, or chart.
- **OnChange** - How the app responds when the user changes the value of a control. For example, when a user selects a different value in a Dropdown control.
- **OnSelect** - How the app responds when the user taps or clicks a control.
- **Reset** - Whether a control reverts to its default value. For more information, see **Reset function in Power Apps**<sup>44</sup>.
- **Text** - Text that appears on a control or that the user types into a control.
- **Tooltip** - Explanatory text that appears when the user hovers over a control.
- **Visible** - Whether a control appears or is hidden.

---

<sup>44</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/functions/function-reset>

To view a list of all the controls and their properties, see [Controls and properties in Power Apps<sup>45</sup>](#).

## Entering and displaying data with text controls

In Power Apps, text controls are used for all kinds of purposes, such as displaying text, numbers, dates, and calculating currency.

For example, you could calculate the value of two Text input controls and display the results in a Label control.

There are several text controls that you could incorporate into your app. Each of these controls have slightly different purposes and use cases.

For example, maybe you're creating an employee survey app, and you want to get employee feedback. In this scenario, you would use the **Text input** control and modify the **Mode** property to be **Multiline** instead of single line. Most likely, every app that you work with will have text controls so familiarizing yourself with these controls will be helpful as you develop your app. The follow list details the different text controls that are available.

- **Label** - A label shows data that you specify as a literal string of text, which appears exactly the way you type it, or as a formula that evaluates to a string of text. Labels often appear outside of any other control (such as a banner that identifies a screen), as a label that identifies another control (such as a rating or audio control), or in a gallery to show a specific type of information about an item.
- **Text input** - A box in which the user can type text, numbers, and other data. For example, a user can specify data by simply typing in a text input control. Depending on how you configure the app, that data might be added to a data source, used to calculate a temporary value, or incorporated in some other way.
- **HTML text** - An HTML text control not only shows plain text and numbers but also converts HTML tags, such as non-breaking spaces.
- **Rich text editor** - The rich text editor control provides the app user a WYSIWYG editing area for formatting text. This control should be used if you want to allow the user to provide numbered lists or bullet lists. A good example is a Power Apps app that used to collect content for an article or newsletter, where you allow the user to add formatted text that would be helpful for the person compiling the article.
- **Pen input** - A control in which the user can draw, erase, and highlight areas of an image. The user can use this control like a whiteboard, drawing diagrams and writing words that can be converted to typed text.

Here's a closer look at the Label control and a few examples so you can get a better idea as to how it works.

<sup>45</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/reference-properties>

First, add a label control to show text.

1. In Power Apps Studio, add a **Label** control.
2. Set **Text** property for the Label to "**Hello, world**" (including the double quotation marks).

Next, create a more dynamic solution by combining a button, gallery, and multiple label controls. In this scenario, you'll create a collection called **CityPopulations** that contains data about the population of various cities in Europe. Next, you'll show that data in a gallery that contains three labels, and you'll specify the type of data that each label will show.

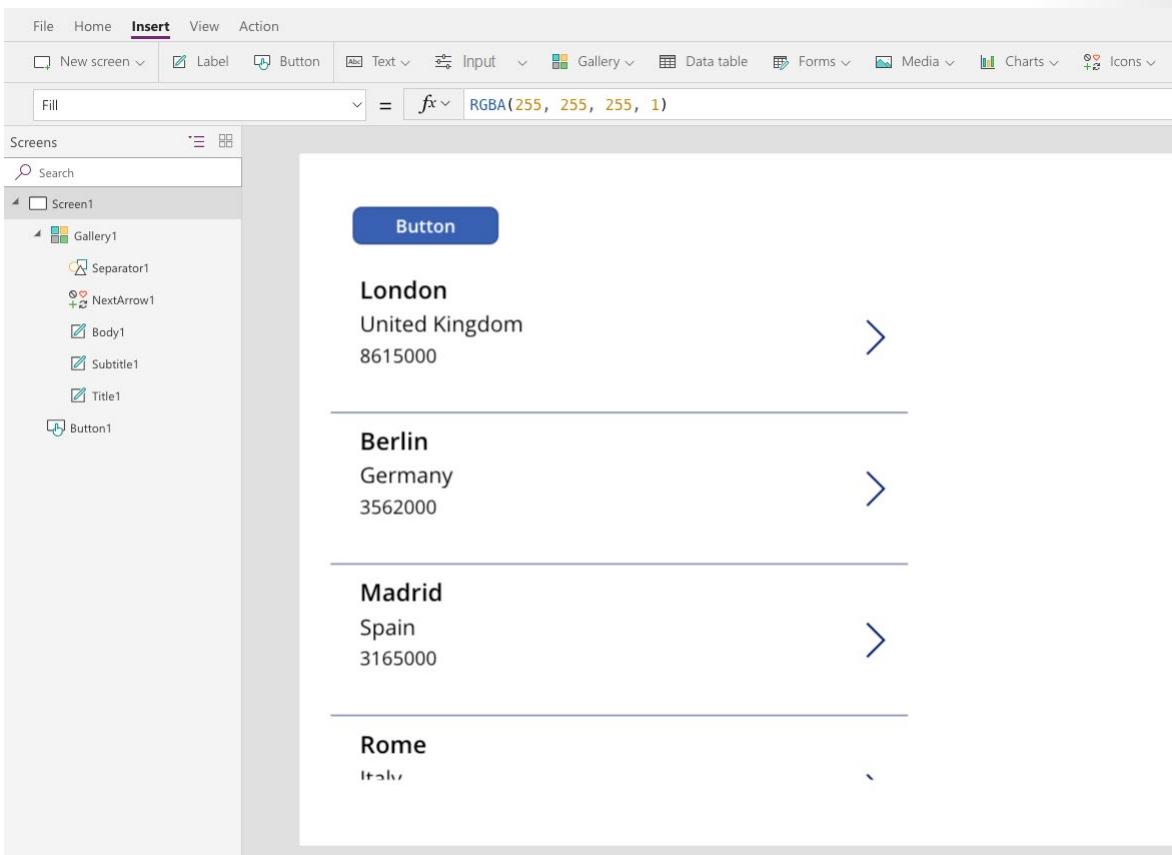
1. Add a button, and set its **OnSelect** property to this formula:

```
ClearCollect(CityPopulations, {City:"London", Country:"United Kingdom", Population:8615000}, {City:"Berlin", Country:"Germany", Population:3562000}, {City:"Madrid", Country:"Spain", Population:3165000}, {City:"Rome", Country:"Italy", Population:2874000}, {City:"Paris", Country:"France", Population:2273000}, {City:"Hamburg", Country:"Germany", Population:1760000}, {City:"Barcelona", Country:"Spain", Population:1602000}, {City:"Munich", Country:"Germany", Population:1494000}, {City:"Milan", Country:"Italy", Population:1344000})
```

2. Press and hold **Alt Key**, and select the **Button** control. (This will create your collection and store all the information.)
3. Add a **Blank vertical** gallery and set its **Items** property to **CityPopulations**.
4. With the gallery selected, in the right pane, change the layout from blank to **Title, subtitle, and body**.
5. Select the top or first Label, in the **Text** property the default for example shows **ThisItem.City**, you could change this to something else if you would like. For more information about the ThisItem operator, see **Operators and data types in Power Apps**<sup>46</sup>.
6. Select the middle or second Label, which shows as **ThisItem.Country**.
7. Select the last or third Label, change the **Text** property to **ThisItem.Population**.

---

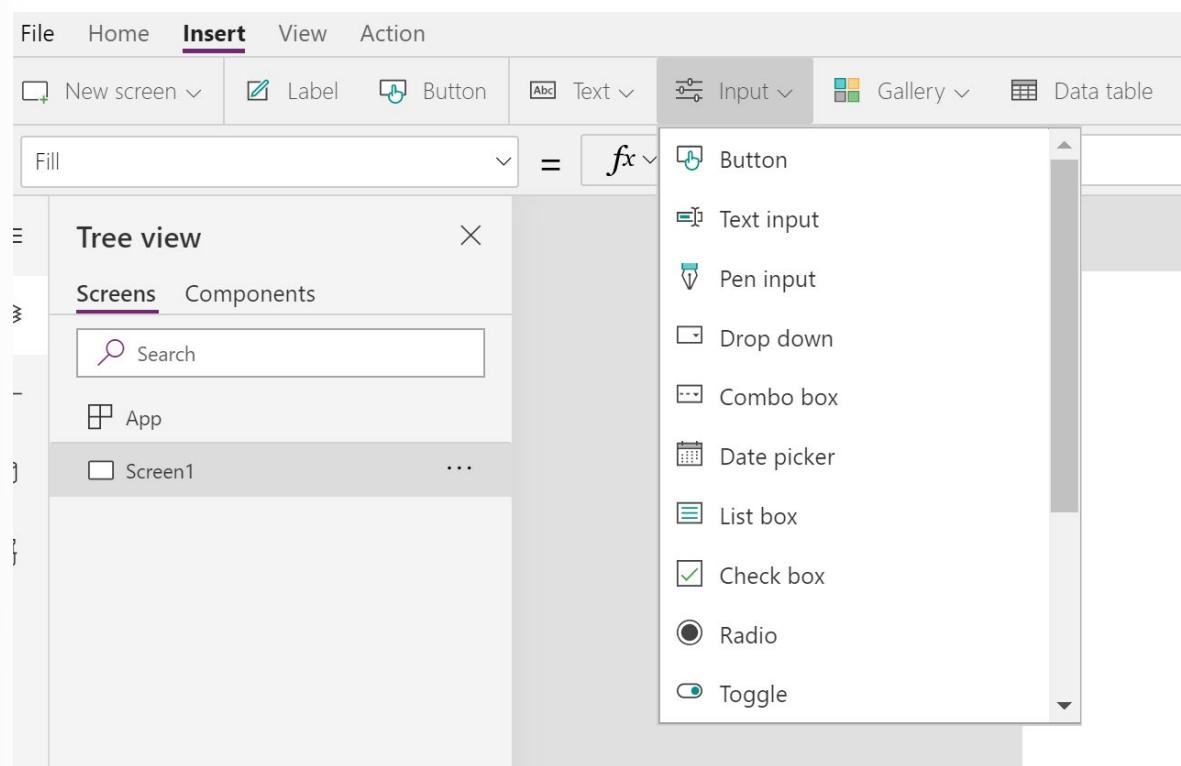
<sup>46</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/functions/operators>



These were just two simple examples demonstrating some basic functionality of a label control. There are many other ways to utilize label controls in your app.

## Additional controls for enhancing your app's usability

At the beginning of the module, you learned what controls are, when to use them, and how to modify control properties. In this section, you get a more detailed look into some of the controls available from the Controls drop-down menu in the ribbon. On the **Insert** tab, if you select the **Input** heading, you will get a list of several different controls.



Remember, every control was designed with different use cases in mind. The following information about a few of the controls will help you to decide when you might use each one.

## Controls with pre-populated values

Each of these controls allows you to determine the values the user can select from. Use these controls when you want to control the values for your data set. This is often an important consideration for reporting.

As a trade-off, you may miss valuable insights that come from free form answers. Here's a closer look at the differences between each of these controls.

- **Drop down** - This control conserves screen real estate, especially when the list contains a large number of choices. The control takes up only one line unless the user selects the chevron to reveal more choices. This control will show a maximum of 500 items.
- **Combo box** - This control allows you to search for items you will select. The search is performed on the server on the **SearchField** property so performance is not affected by very large data sources.
- **List box** - This control always shows all available choices (unlike a **Dropdown**<sup>47</sup> control) and in which the user can choose more than one item at a time

<sup>47</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/controls/control-drop-down>

(unlike a **Radio**<sup>48</sup> control).

- **Check box** - A control that the user can select or clear to set its value to **true** or **false**. The user can specify a Boolean value by using this familiar control.
- **Radio** - A **Radio** control, a standard HTML input control, is best used with only a few, mutually-exclusive options.

## Controls for ratings

When you have a specific need for the app users to rate items, the following two controls provide a better experience than free form text or drop-down controls.

- **Slider** - The user can indicate a value, between a minimum and a maximum value that you specify, by dragging the handle of a slider left-right or up-down, depending on the direction that you choose.
- **Rating** - In this control, the user can indicate, for example, how much they liked something by selecting a certain number of stars.

## Other available controls

- **Toggle** - Use this control to enhance the UI of the app. It functions in the same manner as the check box control.
- **Timer** - A control that can determine how your app responds after a certain amount of time passes. For example, determine how long a control appears or navigate to another screen after a certain amount of time has passed.
- **Button** - Configure the **OnSelect** property of a button control to run one or more formulas when the user clicks or taps the control. The button control is frequently used to submit data to the data source.

## Media

Media controls give the app designer a way to display and share an audio file or video file easily, take a picture, scan bar codes, etc.

For example, you need to build an employee safety training app. For the solution, you want to share a video with your employees and then have them acknowledge that they have completed the training. With the Video control, the safety video can be shown in the app and then the next screen would be a training acknowledgment form. Or maybe you have an expense report app and want to let users upload pictures of their receipts. By using the Camera control, the user can take a photo and attach the picture without leaving the app.

<sup>48</sup> <https://docs.microsoft.com/powerapps/maker/canvas-apps/controls/control-radio>

Here are some details about the Media controls that are available:

- **Video** - This control plays a video clip from a file or from YouTube or Azure Media Services. Closed captions can optionally be shown when specified.
- **Audio** - This control plays a sound clip from a file, a recording from a Microphone control, or the audio track from a video file.
- **Camera** - A control which allows the user to take photos by using the camera on the device.
- **Bar code scanner** - This control opens a native scanner on an Android or iOS device. The scanner automatically detects a barcode, QR code, or data-matrix code when in view. The control doesn't support scanning in a web browser.
- **Microphone** - A control that allows app users to record sounds from their device as long as the device has a microphone accessible. Audio is stored in 3gp format in Android, AAC format in iOS, and OGG format in web browsers.
- **Add picture** - With this control users can take photos or upload image files from their device and update the data source with this content. On a mobile device, the user is presented with the device's choice dialog to choose between taking a photo or selecting one that is already available.

Here's how you can add a video to your app. In this example, you will have two screens. When a user selects the button on the first screen they will navigate to the second screen where a video will start playing automatically.

1. In Power Apps studio, add a new blank screen. You should now have two screens.
2. Make sure that you are on the first screen and add a **Button** control.
3. On the left, under **Screens**, select **Screen 2**.
4. On the ribbon, add a **Video** control.
5. For the **Media** property, in the formula box, enter a video URL from YouTube. The URL must be inside quotations.
6. Now set the **AutoStart** property to **true**.
7. Go back to Screen 1, set the **OnSelect** property for the button to:  
**Navigate(Screen2,ScreenTransition.Cover)**.

Test out your app. Put the app in preview or play mode and press the button.

## Summary

In this module, you learned more about controls and how you can incorporate them into your app. You also learned about the properties of controls and how you can modify them to enhance the functionality

of your app. Depending on the app that you're developing, you can use several different controls, or you can only use a few specific ones. Here are a few important points to remember:

- There are many control options for placing text on a screen or allowing a user to enter text.
- Use the pre-populated value controls to provide choices for the user.
- By adding media controls directly to your app, you can enhance the functionality and reduce the number of steps for your user. For example, users can enter their expenses and upload their receipts without leaving the app.



## Module 4 Automate a business process using Power Automate

### Get started with Power Automate

#### Introducing Power Automate

Welcome to Power Automate! In this module, you'll learn how to build flows.

If you're a beginner with Power Automate, this module will get you going. If you already have some experience, this module will tie concepts together and fill in the gaps.

#### Learning objectives

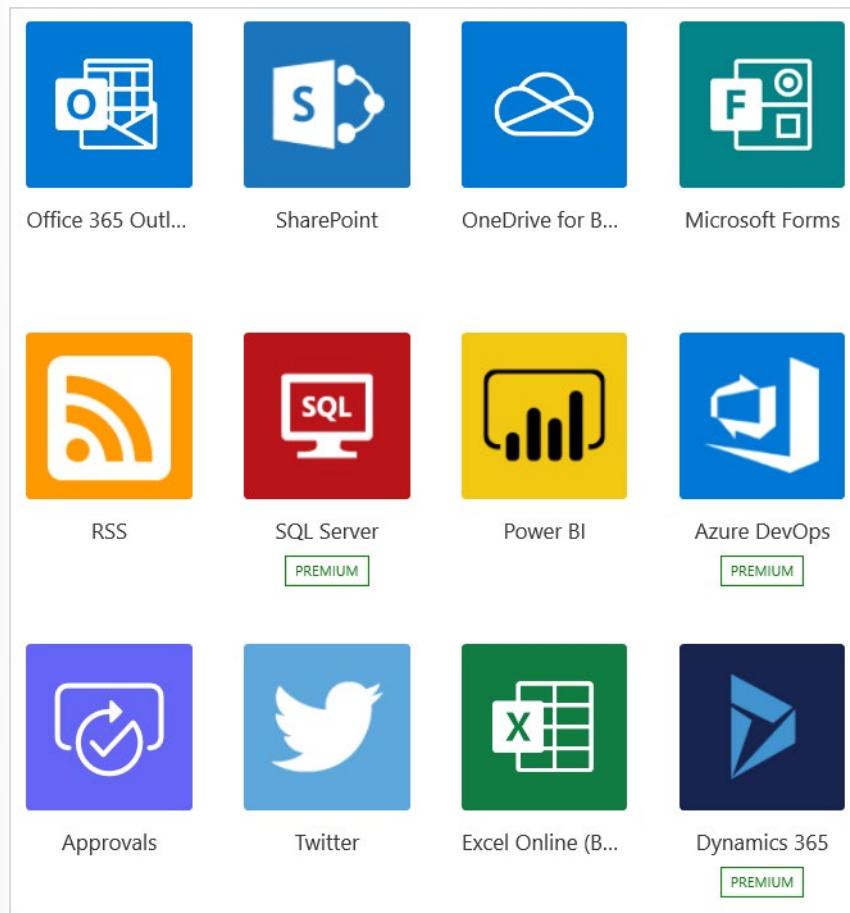
In this module, you will:

- Learn what Power Automate is and how it can be used
- Create a flow that automatically saves email attachments
- Learn how to create a button flow to send yourself a reminder
- Create a flow that sends you notifications
- Create a flow that copies files
- Create a flow that runs on a schedule
- Create a flow that posts tweets
- Create a flow that your team can use

#### What is Power Automate?

Power Automate is an online workflow service that automates actions across the most common apps and services. For example, you can create a flow that adds a lead to Microsoft Dynamics 365 and a record in MailChimp whenever someone with more than 100 followers tweets about your company.

When you sign up, you can connect to more than 220 services, and can manage data either in the cloud or in on-premises sources like SharePoint and Microsoft SQL Server. The list of applications you can use with Power Automate grows constantly.



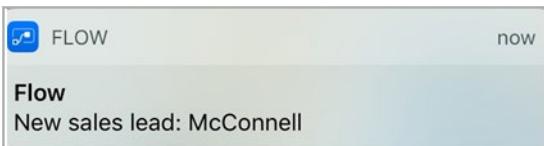
## What can you do with Power Automate?

You can use Power Automate to automate workflows between your favorite applications and services, sync files, get notifications, collect data, and much more.

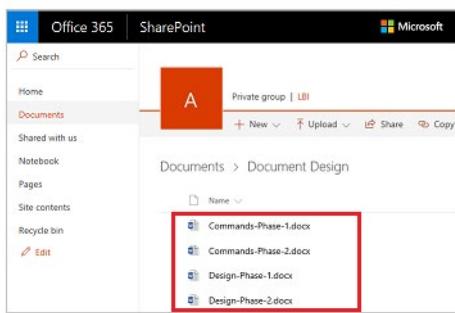
For example, you can automate these tasks:

- Instantly respond to high-priority notifications or emails.
- Capture, track, and follow up with new sales leads.
- Copy all email attachments to your OneDrive for Business account.
- Collect data about your business, and share that information with your team.
- Automate approval workflows.

A common use of Power Automate is to receive notifications. For example, you can instantly receive an email or a push notification on your phone whenever a sales lead is added to Dynamics 365 or Salesforce.



You can also use Power Automate to copy files. For example, you can ensure that any file that's added to Dropbox is automatically copied to SharePoint, where your team can find it.

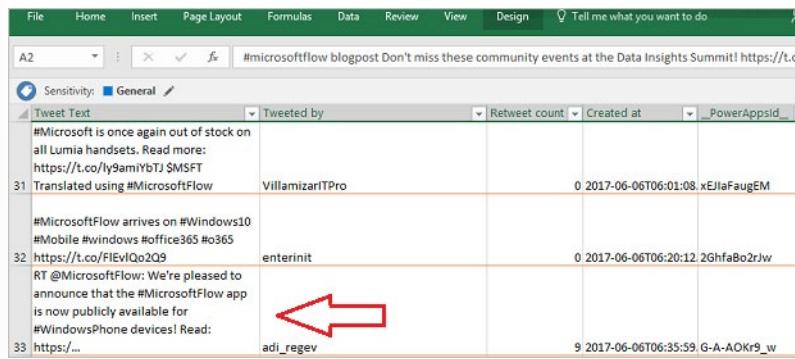


You can monitor what people are saying about your business by creating a flow that runs whenever someone sends a tweet with a certain hashtag. The flow can add details about each tweet to a Facebook post, a SQL Server database, a SharePoint list, or even a Microsoft Excel file that's hosted on OneDrive for Business--whichever service works for you.

You can create actions to connect the data you collect to Microsoft Power BI, spot trends in that data, and ask questions about it.

The following example shows a flow that saves tweets with the hashtag #MicrosoftFlow to an Excel file.

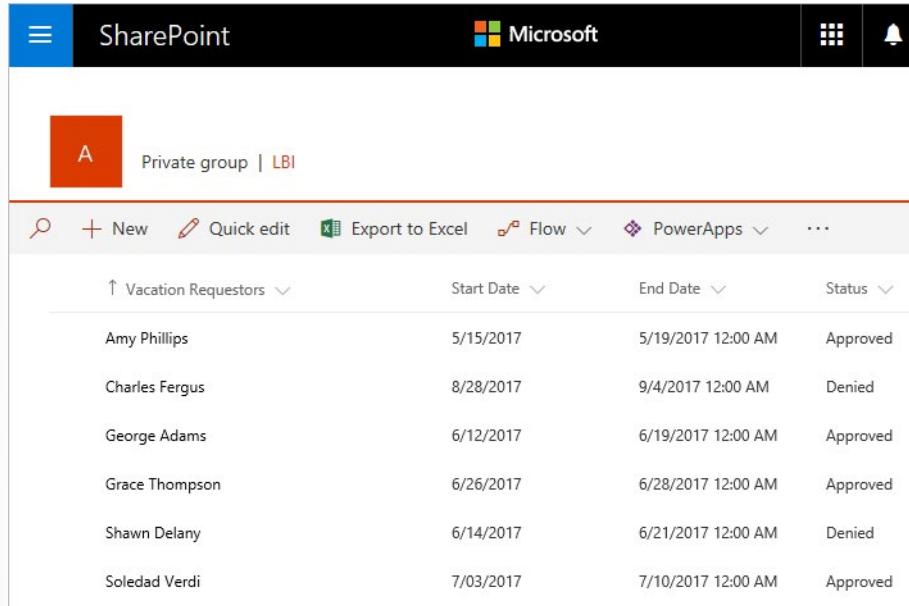




A screenshot of Microsoft Excel showing a table of tweets from a SharePoint list. The table has columns for Tweet Text, Tweeted by, Retweet count, Created at, and PowerAppsid. A red arrow points to the 'Created at' column.

|    | Sensitivity:                                                                                                                                                                                                                                   | General         | Tweet Text | Tweeted by                       | Retweet count | Created at | PowerAppsid |
|----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------|------------|----------------------------------|---------------|------------|-------------|
| 31 | #Microsoft is once again out of stock on all Lumia handsets. Read more: https://t.co/y9amIyBTJ \$MSFT Translated using #MicrosoftFlow                                                                                                          | VillamizarITPro | 0          | 2017-06-06T06:01:08. xEJlaFaugEM |               |            |             |
| 32 | #MicrosoftFlow arrives on #Windows10 #Mobile #windows #Office365 #o365 https://t.co/FIevlQz209 RT @MicrosoftFlow: We're pleased to announce that the #MicrosoftFlow app is now publicly available for #WindowsPhone devices! Read: https://... | enterinit       | 0          | 2017-06-06T06:20:12. 2GhfaBo2rJw |               |            |             |
| 33 | RT @MicrosoftFlow: We're pleased to announce that the #MicrosoftFlow app is now publicly available for #WindowsPhone devices! Read: https://...                                                                                                | adi_regev       | 9          | 2017-06-06T06:35:59. G-A-AOKr9_w |               |            |             |

Also, you can automate approval loops for things like vacation requests on a SharePoint list.



A screenshot of a SharePoint list titled 'Vacation Requestors'. The list shows six rows of data with columns for 'Vacation Requestors', 'Start Date', 'End Date', and 'Status'. The 'Status' column shows 'Approved' for most entries except for Charles Fergus, who is 'Denied'. A red arrow points to the 'Status' column.

| Vacation Requestors | Start Date | End Date           | Status   |
|---------------------|------------|--------------------|----------|
| Amy Phillips        | 5/15/2017  | 5/19/2017 12:00 AM | Approved |
| Charles Fergus      | 8/28/2017  | 9/4/2017 12:00 AM  | Denied   |
| George Adams        | 6/12/2017  | 6/19/2017 12:00 AM | Approved |
| Grace Thompson      | 6/26/2017  | 6/28/2017 12:00 AM | Approved |
| Shawn Delany        | 6/14/2017  | 6/21/2017 12:00 AM | Denied   |
| Soledad Verdi       | 7/03/2017  | 7/10/2017 12:00 AM | Approved |

For more ideas, browse our list of templates. Templates help you build flows by making a few configuration changes. For example, you can use templates to easily build flows to send yourself weather forecasts, reminders at regular intervals, or phone notifications whenever your manager sends you mail.

|                                                                                    |                                                                         |              |           |        |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------|--------------|-----------|--------|
|   | <b>Send myself a reminder in 10 minutes</b>                             | By Microsoft | Instant   | 303832 |
|  | <b>Get today's weather forecast for my current location</b>             | By Microsoft | Instant   | 216639 |
|   | <b>Get a push notification when you receive an email from your boss</b> | By Microsoft | Automated | 181614 |
|  | <b>Send a customized email when a new SharePoint list item is added</b> | By Microsoft | Automated | 144443 |

Have an idea for a flow that you don't see in the list? Create your own from scratch and, if you want, share it with the community!

## Where can I create and administer a flow?

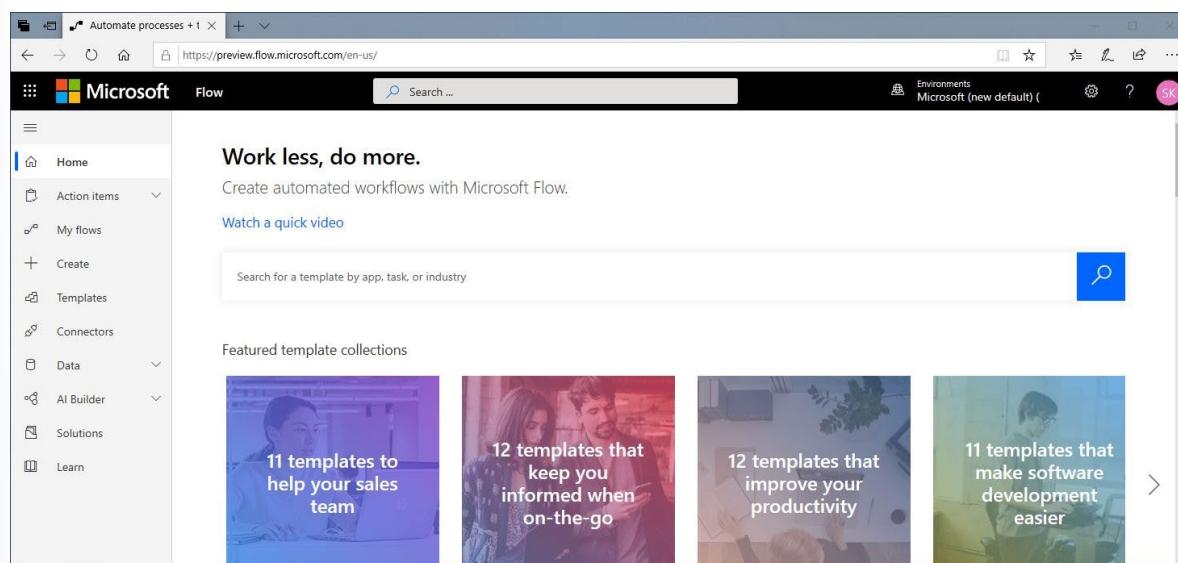
You can create a flow and perform administrative tasks in a browser or, if you download the Power Automate mobile app, on your phone.

Here are some of the tasks you can perform with the mobile app:

- Turn flows on or off from wherever you are.
- See when a flow has failed.
- Review detailed run history reports.
- View and filter runs by notification type.

## A brief tour of Power Automate

Let's jump into Power Automate, and we'll show you around. We have tons of information for you to learn about how to use Power Automate.



When you sign in to Power Automate, you'll find these menus:

- **Action items**, where you can manage approvals and business process flows.
- **My flows**, where your flows reside.
- **Create**, where you start a new flow.
- **Templates**, where you can take a look at some of the most popular templates. These should give you some great ideas for flows you want to try.
- **Connectors**, where you can connect from one service to another.
- **Data**, where you can access tables, connections, custom connectors and gateways.
- **Solutions**, where you can manage your solutions.
- **Learn**, where you can find information that will help you quickly ramp up on Power Automate.

For now, let's focus on the ? menu next to your login, which has these options:

- **Documentation** is where our advanced topics reside. If you want to really understand a feature or function, you can do a deep dive here to figure things out.
- **Learn** has learning paths to guide you through using Power Automate, all the way from beginning techniques to advanced scenarios.
- **Support** is a great landing place to find help.
- **Roadmap** is where you can get a glimpse into what will be made in the next product update.
- **Community** is a place to plug into and find out how other people use Power Automate.
- **Give Feedback** taps into a community of power users, and is where you can send comments and questions to developers and other experienced users.
- **Blog** keeps you up to date about the most recent developments and releases in the Power Automate ecosystem.
- **Pricing** can help you choose the right plan for you or your business.

## What's next?

Now that you have a taste of what Power Automate is and what it can do, let's take a look at what makes a flow.

## Exercise - Create your first flow

In this unit, you'll see more of Power Automate as you build your first flow.

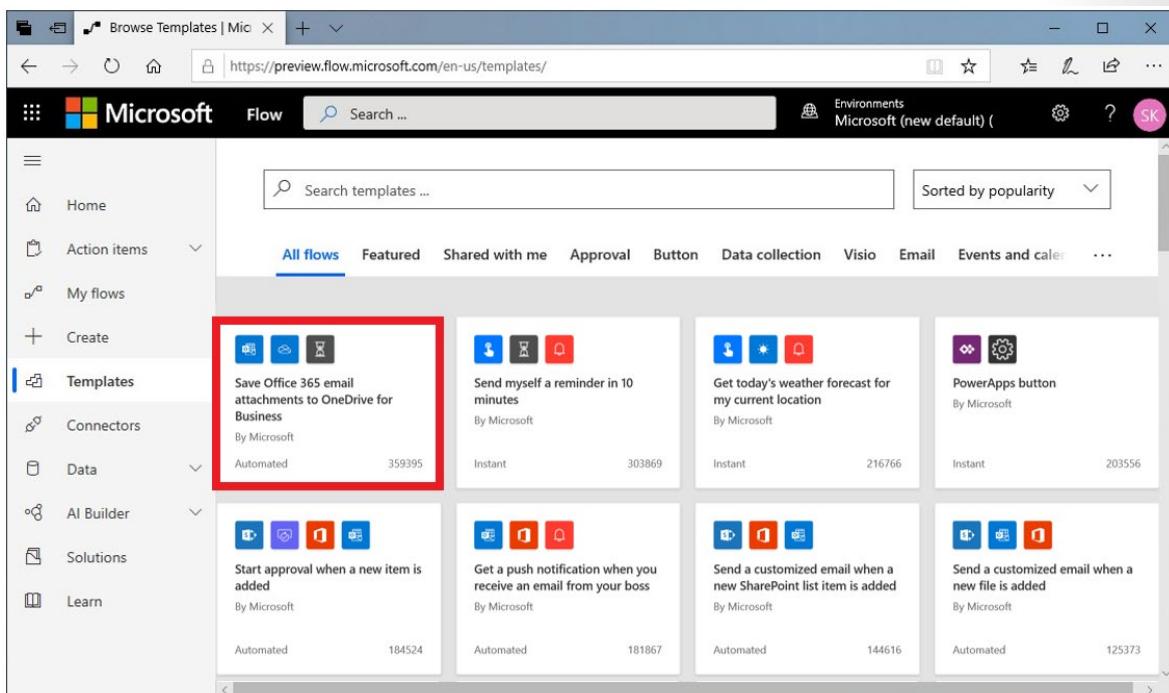
It can be time consuming to search for attachments through email. The flow that you'll build saves time by storing all your email attachments in a folder on your Microsoft OneDrive for Business account.

## Choose a template

Our many templates will get your flows flowing quickly. They'll help you connect the services you're already using in more meaningful ways.

Sign in to **Power Automate**<sup>1</sup>, and select the **Templates** menu. You can sign up for Power Automate with a Microsoft account.

Select the **Save Office 365 email attachments to OneDrive for Business** template.



## Create the flow

**Save Office 365 email attachments to OneDrive for Business** is one of our one-click templates, in which you can answer questions that are necessary to build the flow, so that you don't have to write a line of code.

On the template graphic, there's a description of what the template does and what it needs to succeed.

<sup>1</sup> <https://ms.flow.microsoft.com>

You'll be asked to provide credentials for the Microsoft Office 365 Outlook and Microsoft OneDrive for Business services. If you regularly use both services, you'll already be signed in.

1. Click on the template and select **Create Flow**.
2. On the next page, Power Automate creates the flow for you.
  - It will connect to your work email to get any attachments.
  - It will then create a folder on your OneDrive for Business account to automatically put every attachment that's sent to your work email address in that folder.

The screenshot shows the 'Save Office 365 email attachments to OneDrive for Business' flow details page. At the top, there are various action buttons: Edit, Share, Save As, Delete, Send a copy, Submit as template, Export, Analytics, Turn off, and Repair tips off. Below the header, the flow name is 'Save Office 365 email attachments to OneDrive for Business'. The 'Details' section contains the flow description: 'Now you can have easy access to your Office 365 email attachments from OneDrive for Business. All attachments sent to your Office 365 inbox will be saved in a folder called Email attachments from Flow.' It also lists the owner as 'Sari' and provides status information: Status (On), Created (Oct 15, 09:39 AM), Modified (Oct 15, 09:39 AM), Type (Automated), and Plan (Per-user plan). The 'Connections' section shows two connections: 'Office 365 Outlook' and 'OneDrive for Business'. The 'Owners' section shows 'Sari' as the owner. The 'Runs' section indicates that the flow has not run yet.

3. Select the **My flows** menu.
4. Select the flow you just created and click **Edit** to see how it works.

The screenshot shows the 'Flows' page. At the top, there is a navigation bar with 'New', 'Edit' (highlighted with a red arrow), 'Share', 'Save As', and a search bar. Below the navigation bar, the title 'Flows' is displayed. Underneath, there are three tabs: 'My flows' (selected), 'Team flows', and 'Business process flows'. The main area displays a list of flows. The first flow listed is 'Save Office 365 email attachments to OneDrive for ...', which was created 20 minutes ago and is of type 'Autom...'. The flow icon includes a checkmark and a blue square with a white icon.

5. Send an email with an attachment, or have another user send an email with an attachment. You then should see a green check mark, which indicates that the flow succeeded.
6. Select **Edit** to see how the flow is defined.
7. Select **Succeeded** to see the run history and the results.

The screenshot shows the 'Details' section of a Power Automate flow named 'Save Office 365 email attachments to OneDrive for Business'. The flow is currently 'On' and was created on Oct 15, 09:39 AM. It was modified on Oct 15, 10:18 AM. The type is 'Automated' and the plan is 'Per-user plan'. The owner is Sari. Under 'Connections', 'Office 365 Outlook Permissions' and 'OneDrive for Business Permissions' are listed. The 'Runs' section shows four successful runs: one on Oct 16 at 01:29 PM (1 d ago), three on Oct 15 at 12:41 PM, 12:10 PM, and 10:30 AM (all 2 d ago). Each run duration is 0:00:03, 0:00:02, 0:00:02, and 0:00:01 respectively.

In this case, all parts of the flow were successful.

The screenshot shows the flow editor with the following sequence of steps:

- On new email**: Trigger step, completed successfully (0s).
- Apply to each Attachment on the email**: Action step, completed successfully (3s).
- Create file**: Sub-action step under 'Apply to each Attachment on the email', completed successfully (3s).
- Condition**: Decision step, currently inactive (0s).

A message bar at the top says: "Your flow ran successfully."

## Important concepts in Power Automate

Keep these concepts in mind when building flows:

- Every flow has two main parts: a *trigger*, and one or more *actions*.
- You can think of the trigger as the starting action for the flow. The trigger can be something like a new email arriving in your inbox or a new item being added to a SharePoint list.
- Actions are what you want to happen when a trigger is invoked. For example, the new email trigger will start the action of creating a new file on OneDrive for Business. Other examples of actions include sending an email, posting a tweet, and starting an approval.

These concepts will come into play later, when you build your own flows from scratch. In the next unit, we'll look at the Power Automate mobile app and its capabilities.

## Exercise - Learn to use the Power Automate mobile app

Of course we have an app – the Power Automate mobile app! From this app, you can access these features:

- Activity Feed
- Browsing
- Buttons
- Managing Flows

First, you'll need to download and install the Power Automate mobile app from your app store.

After it's installed, start it and sign in.

When you first start the app, you'll see the Activity Feed. The Activity Feed is the place to see what's happening with your flows. It won't be the full experience you'd expect from your PC, but it will show you useful details.

For example, you'll see a flow's last activity. You can see whether the flow succeeded or failed. If it failed, you'll see which step it failed on.

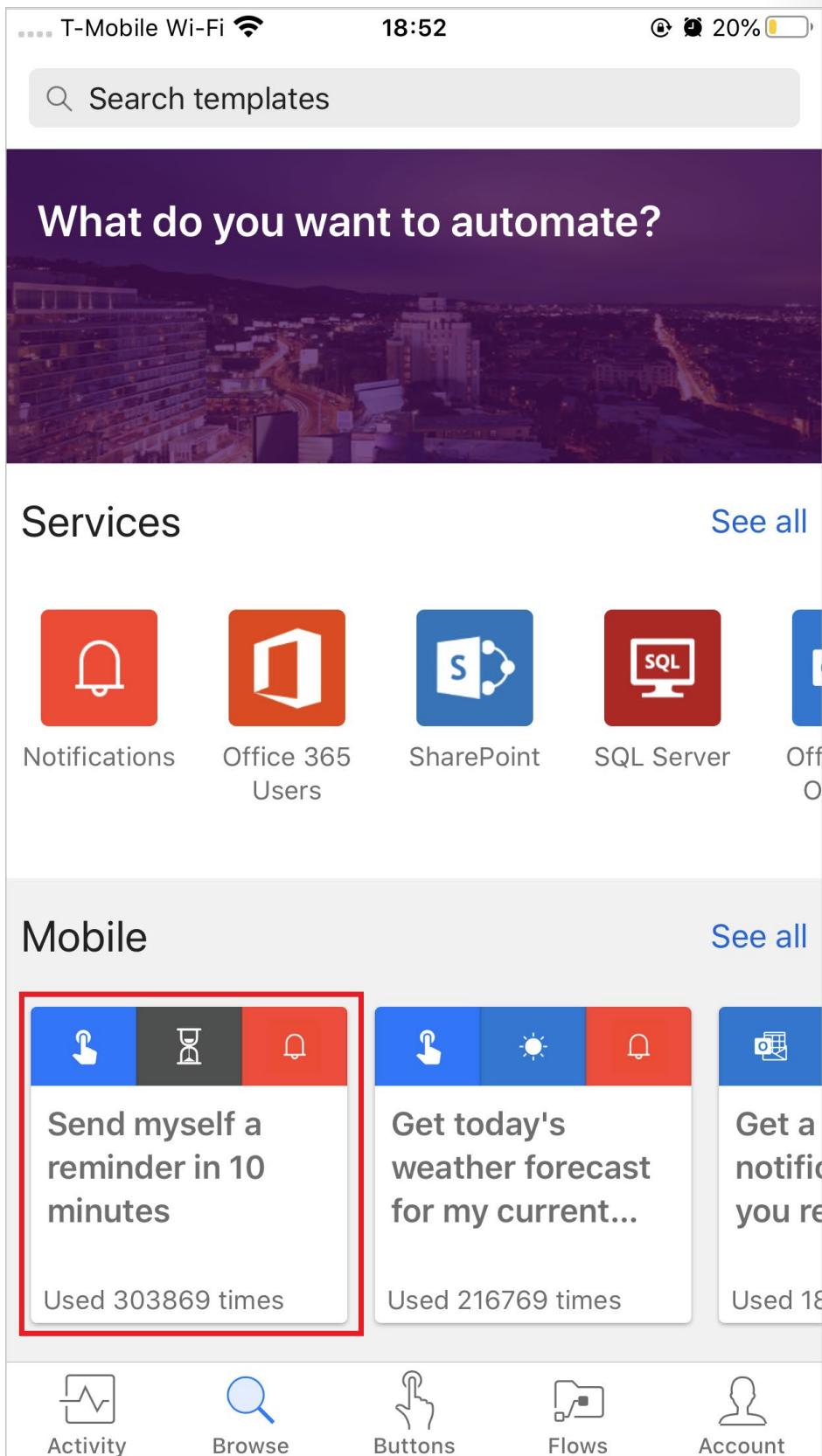
## How button flows are started

Buttons are flows that are started through a manual action. For example, you can create a button to send a *Working from home today* email to your manager. If you live far from your workplace, you can then use this button on days when the traffic is a mess!

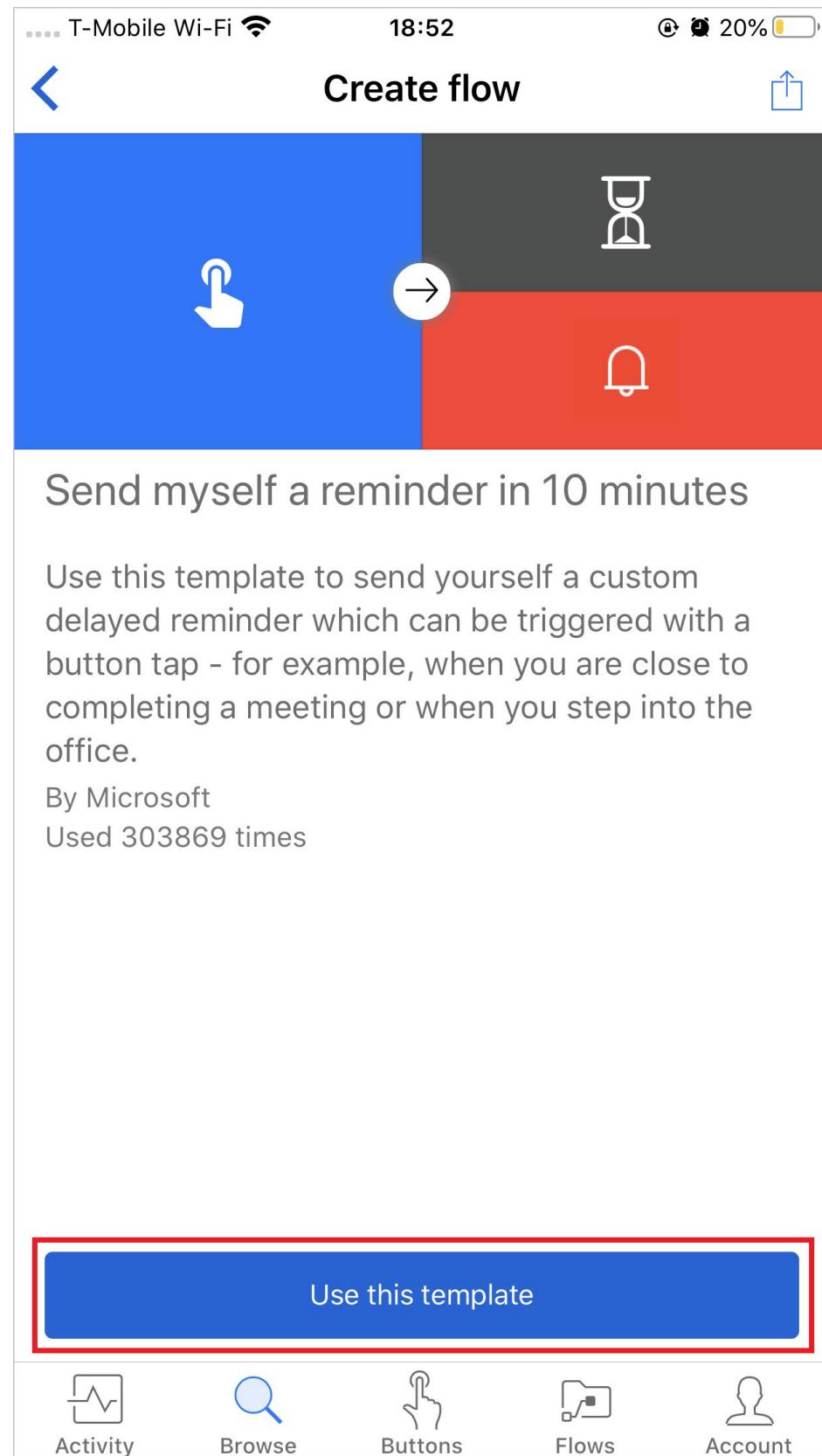
- Select **Buttons** to use some of these flows.
- Select **Browse** to check out templates for more button flows that you can add to your collection.

To show you how you can use buttons, we'll use the **Send myself a reminder in 10 minutes button** template.

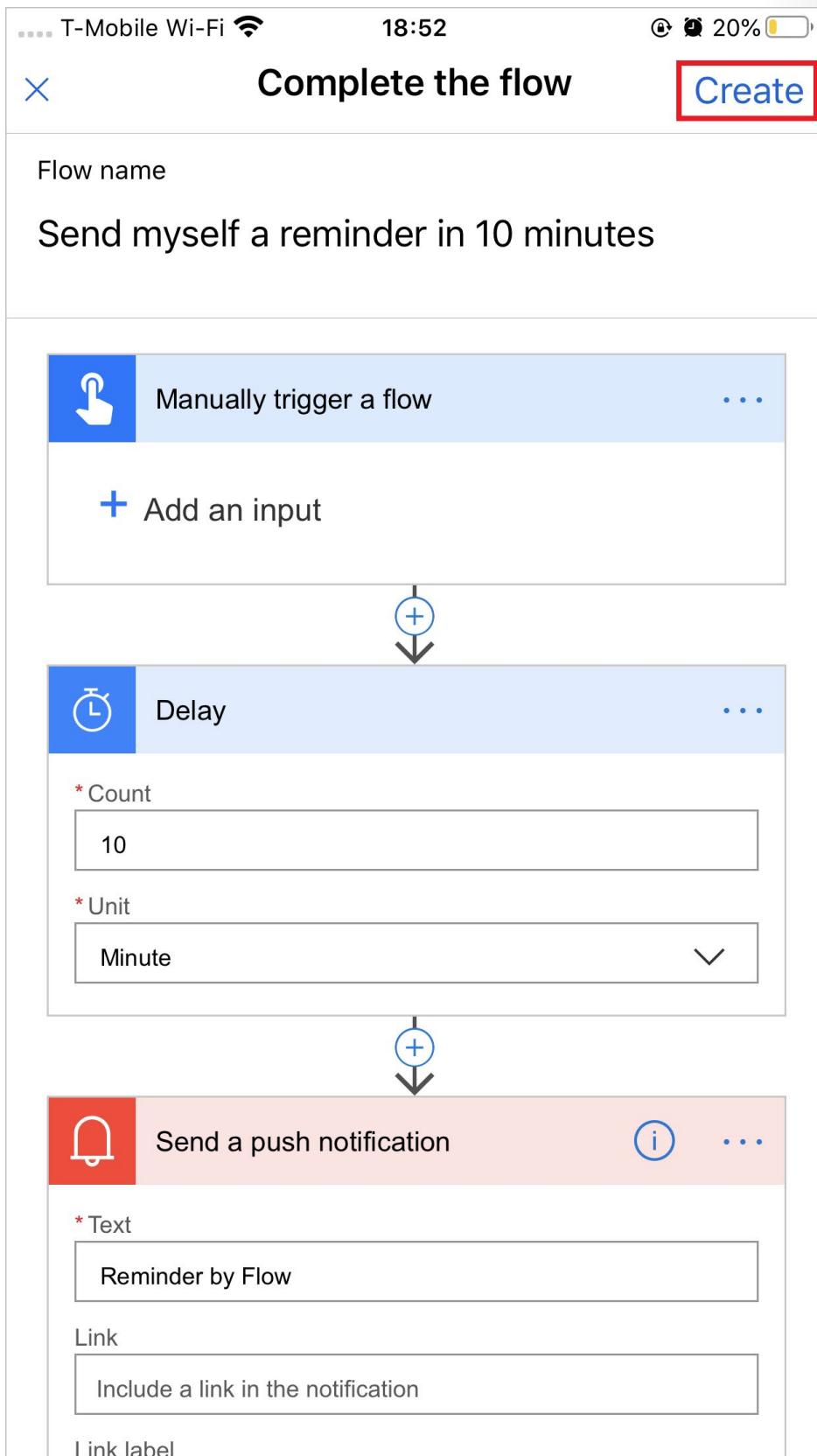
1. Select **Browse**.
2. Select the **Send myself a reminder in 10 minutes button** flow.



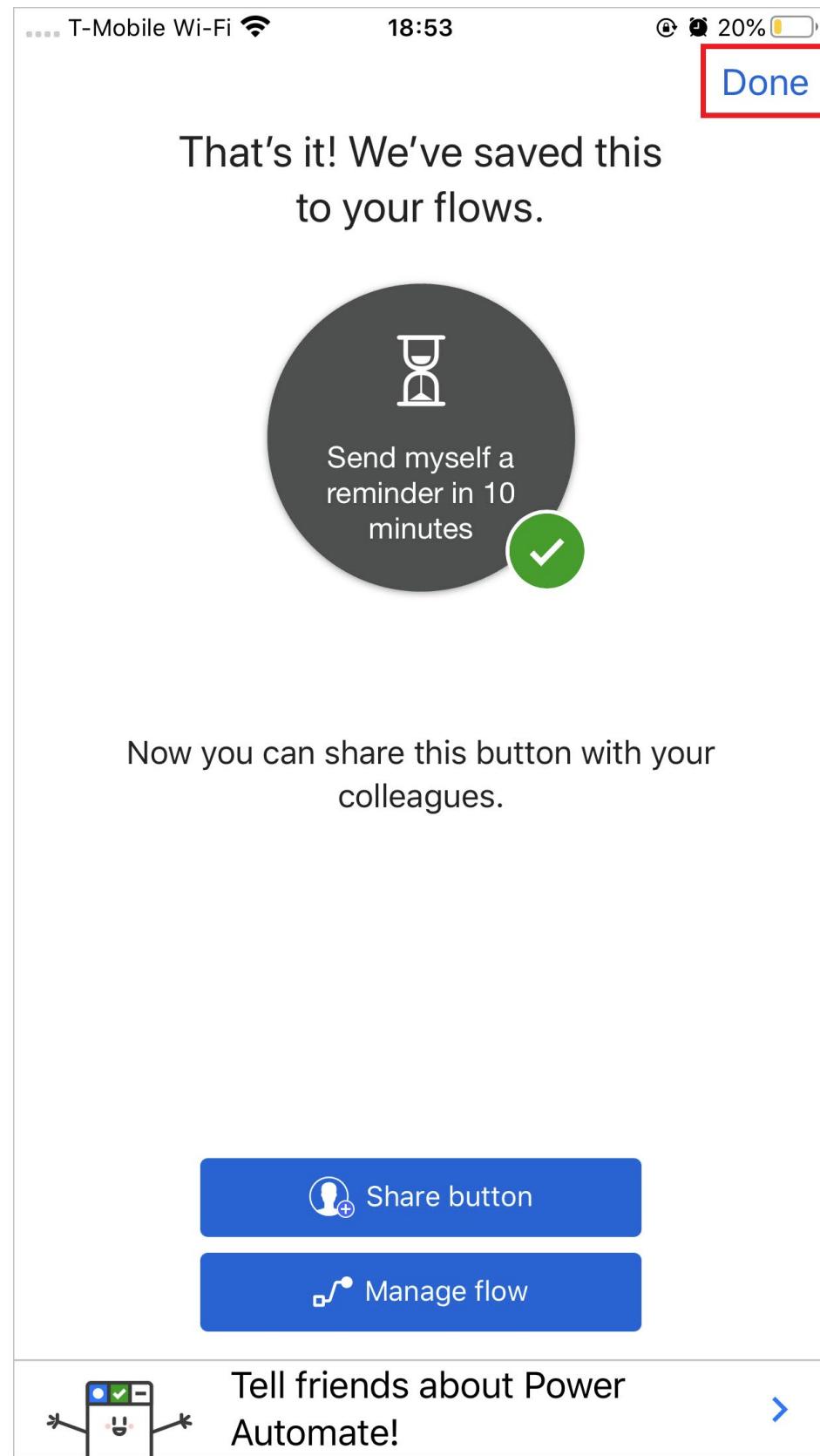
3. Select **Use this template**.



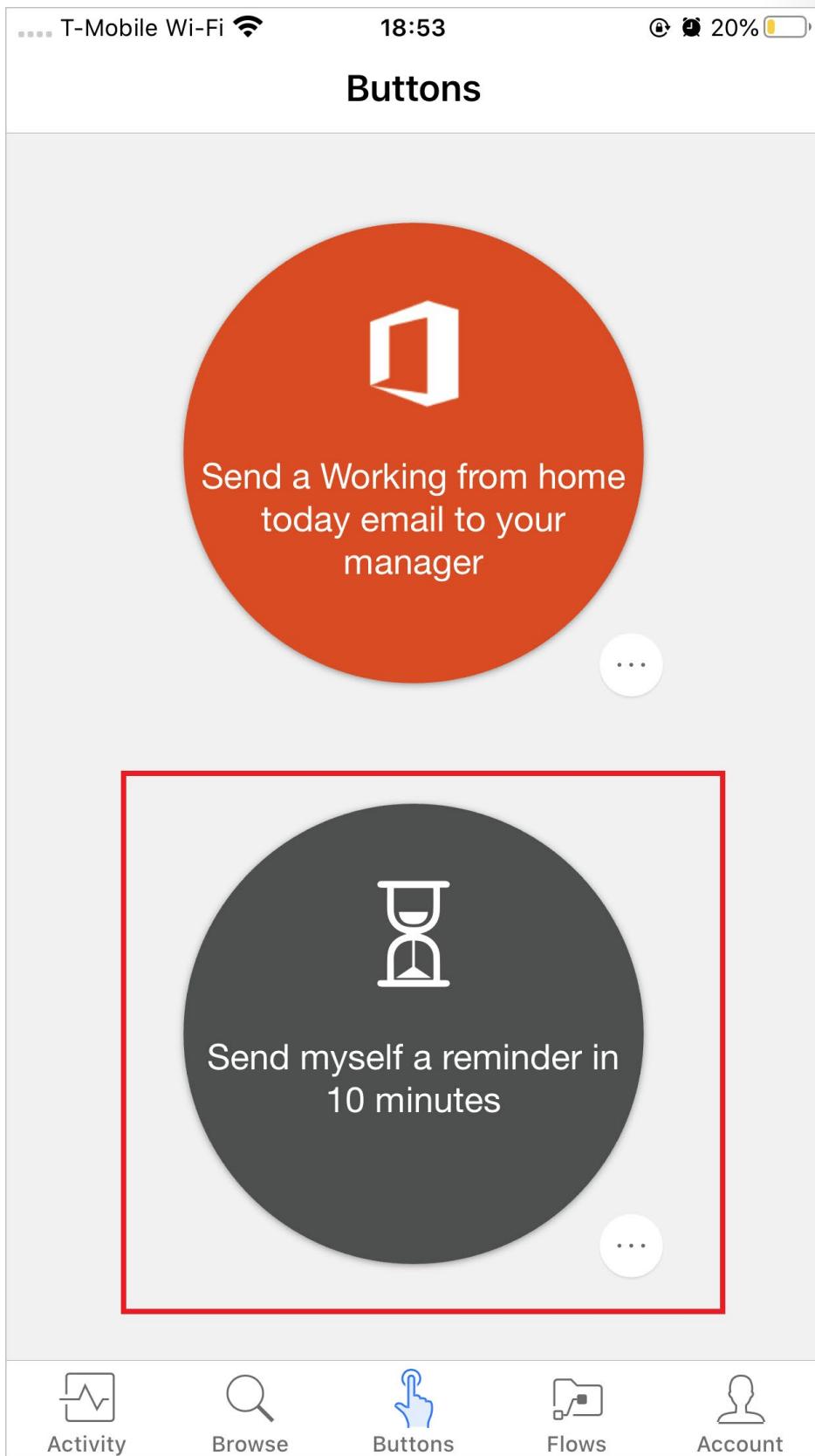
4. Select **Create**, and then select **Done**.



The flow is saved.



5. Select **Buttons** to see the new flow.



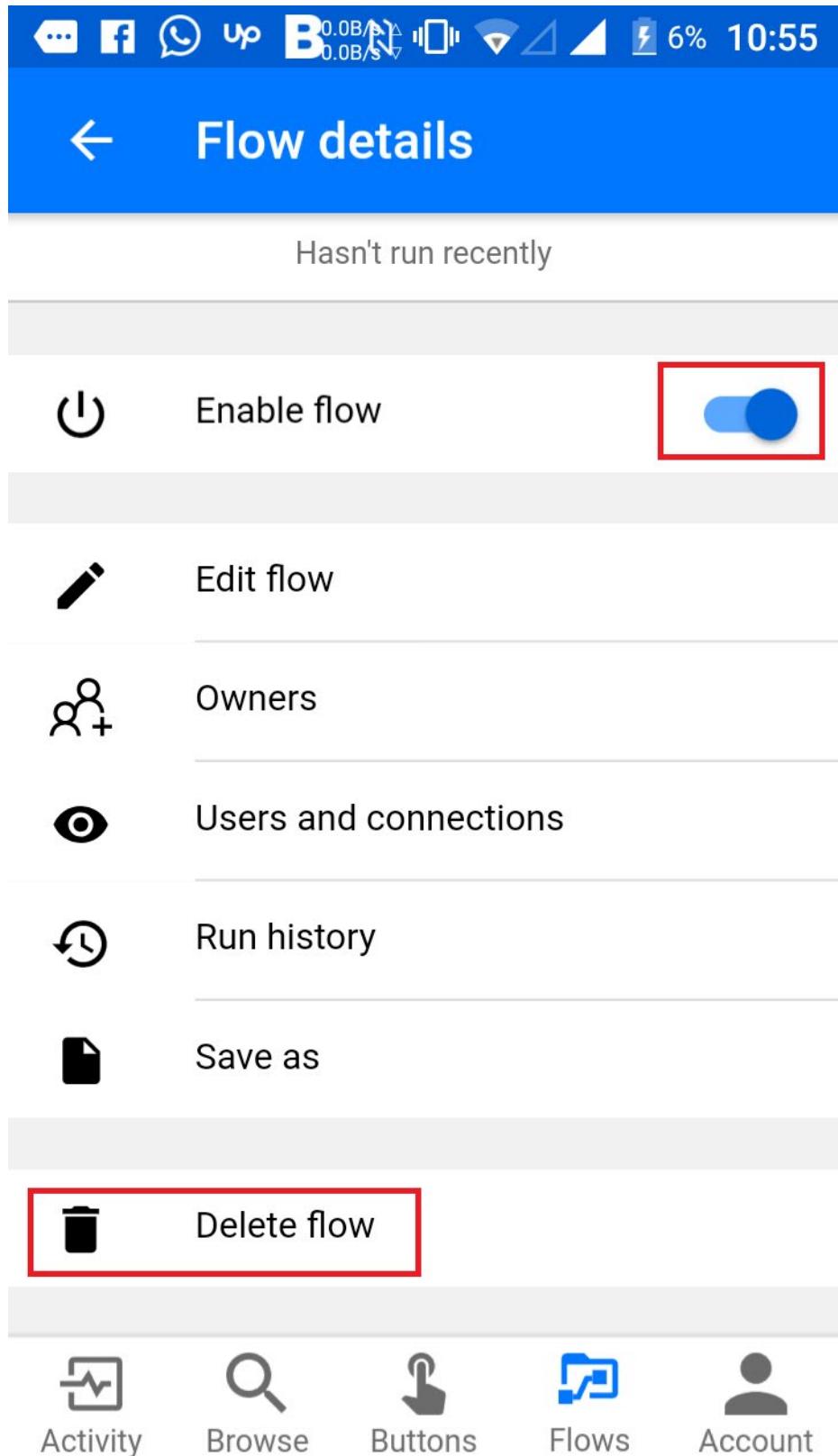
6. Select the flow. In 10 minutes, you'll get a reminder.

It's simple to add more buttons to your collection.

## Modify or delete a flow

If you want to change or delete one of your flows, it's easy.

1. Select **Flows**.
2. Select one of your flows.
3. Select one of the options:
  - To enable or disable the flow, toggle the **Enable flow** option on or off.
  - To change the flow, select **Edit flow**.
  - To get an idea of the successful and unsuccessful runs of the flow, select **Run history** to view the history of the flow.
  - To delete a flow, select **Delete flow**.



The next unit shows how to receive text and email notifications from flows.

# Exercise - Receive text and email notifications from flows

A common use of Power Automate is to get a notification when something happens. Notifications can be emails, text messages, or push notifications on your phone.

In this unit, you'll create a flow that generates a push notification whenever you receive an email from your manager.

## Get the mobile app

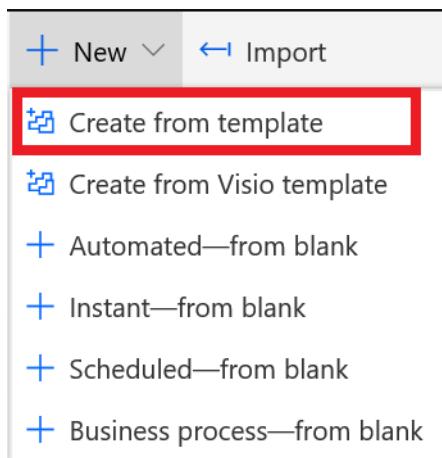
It's easy to create a flow that sends push notifications to your mobile device. Keep in mind that you'll need the Power Automate mobile app to receive push notifications. The mobile app is available for **Google Android<sup>2</sup>** and **Apple iOS<sup>3</sup>**.

If you're using an unsupported mobile device, consider using Short Message Service (SMS) messages (that is, text messages) instead of push notifications to receive notifications.

## Create a flow that sends push notifications

Power Automate comes with many templates to get you started with creating flows. Let's create a flow by using a template.

1. Sign in to **Power Automate<sup>4</sup>** by using your organizational account.
2. Select **My flows**.
3. Select **New**, and then select **Create from template**.



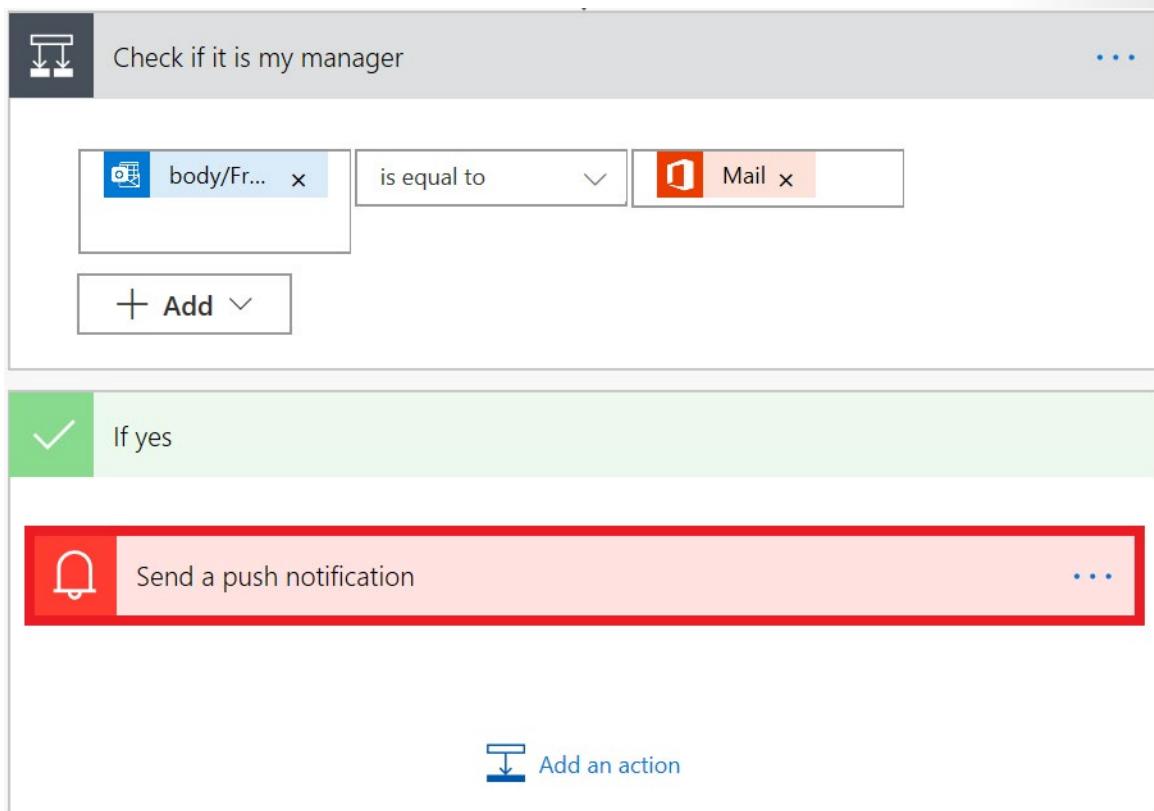
4. Scroll down, and select **Get a push notification when you receive an email from your boss**. You can also quickly find this template by entering *notifications* in the search column.
5. Select **Create Flow**. The flow will automatically render and open to the details and run history page.
6. To edit the flow and see the steps which will be used to get your email profile and your boss's, select **Edit**.

<sup>2</sup> <https://play.google.com/store/apps/details?id=com.microsoft.flow>

<sup>3</sup> <https://itunes.apple.com/app/apple-store/id1094928825>

<sup>4</sup> <https://ms.flow.microsoft.com>

7. Scroll to the bottom of the flow steps to find the **Check if it is my manager** section. Expand it by clicking on the title of that section. Your email address and your manager's are automatically filled in from the profile information that you entered. If your organization does not store your manager information in the active directory, you can manually input your manager's email here. Only administrators have the right to change this information so you cannot edit this portion of your profile data unless you have administrator rights.
8. In the **Send a push notification** section, select the title bar to change the text of the notification that you'll get when an email is received from your boss.



9. To change the text of the email, in the **Text** field, enter a new message. You can also select dynamic content fields in the list.



10. Select **Save** to save and test the flow.

Now, when emails arrive from your boss, you'll get a push notification on your phone.

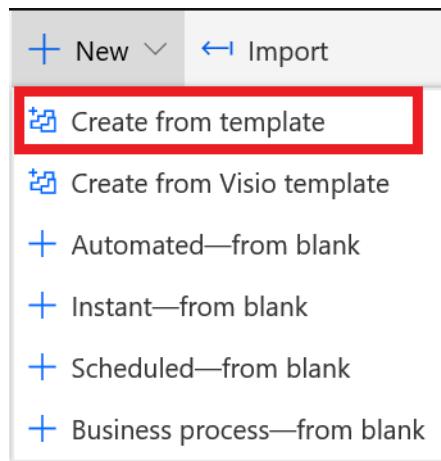
## Exercise - Copy files with flows

At some point, most of us have needed to copy files from one storage service to another. Power Automate makes it easy to automatically move and copy files between two services, like Microsoft OneDrive and Google Drive.

In this unit, you'll use a flow template to copy files from your personal OneDrive to Microsoft OneDrive for Business. Power Automate must have permissions to your OneDrive folders to do this.

### Create a flow that copies files

1. Sign in to **Power Automate**<sup>5</sup> by using your organizational account.
2. Select **My flows**.
3. Select **New**, and then select **Create from template**.



4. Scroll down, and select **Copy files to OneDrive for Business when they're added to OneDrive**.

You can also quickly find this template by entering *OneDrive for Business* in the search field.

5. Select **Continue**.
6. In the **OneDrive Folder** field, select the folder button.

A screenshot of a configuration screen for a 'OneDrive Folder'. It has two fields: '\*OneDrive Folder' and '\*OneDrive for Business Folder Path'. The first field contains the placeholder 'The unique identifier of the folder.' with a small folder icon at the end, which is highlighted with a red box. The second field contains the placeholder 'The unique path of the folder.' with a similar folder icon at the end.

7. Select the OneDrive folder that files should be copied from.
8. In the **OneDrive for Business Folder Path** field, select the folder button, and then select the folder that files should be copied to.
9. Select **Create Flow**.
10. To change the flow, select **Edit flow**.

<sup>5</sup> <https://ms.flow.microsoft.com>

Now, whenever a file is put in the selected folder on OneDrive, it will be copied to the selected folder on OneDrive for Business.

## Exercise - Create recurring flows

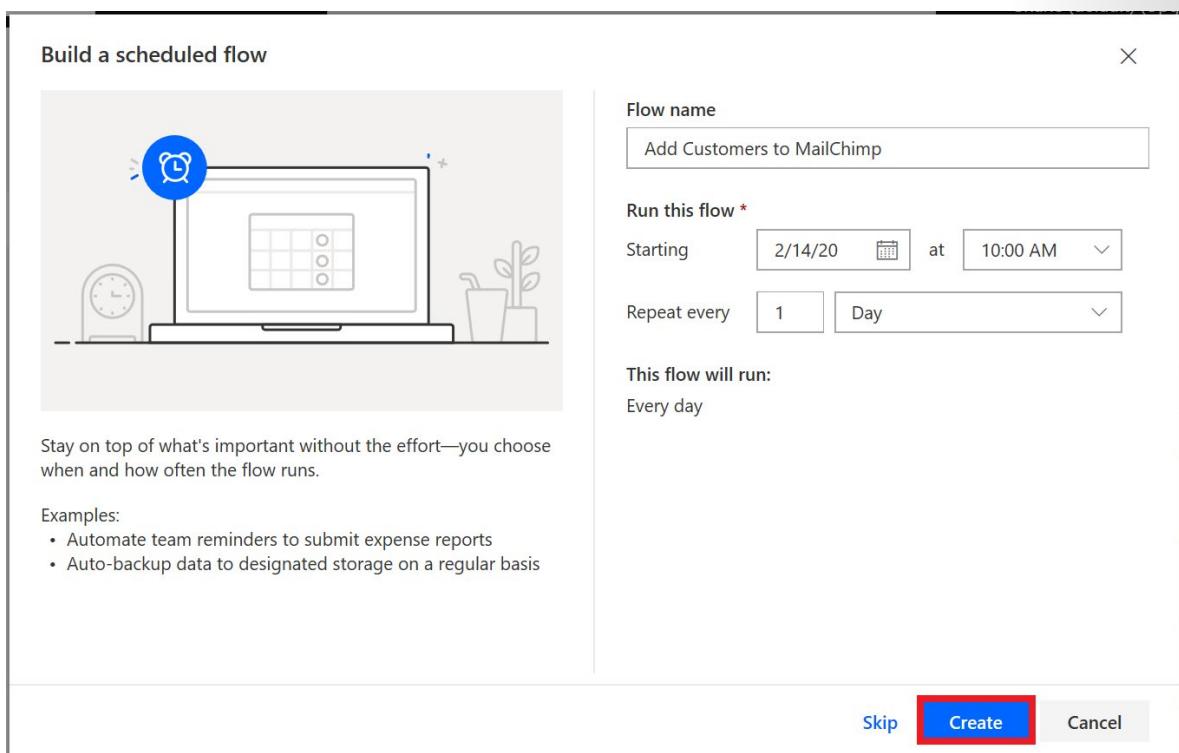
In this unit, you'll learn how to build prescheduled flows by using a trigger called *recurrence*. You'll build a flow for the Contoso marketing team that automatically pulls customer email addresses from a Microsoft Excel workbook on Microsoft OneDrive. You'll then set up the flow so that, once a day, any new email addresses that were added to the workbook are added to a MailChimp customer list.

### Prerequisites

For this scenario, you will need to make an Excel file with a table that contains the following columns: ContactEmail, FirstName, and LastName. Save the Excel file in OneDrive for Business. You'll connect to this file in step 9.

### Create a scheduled flow

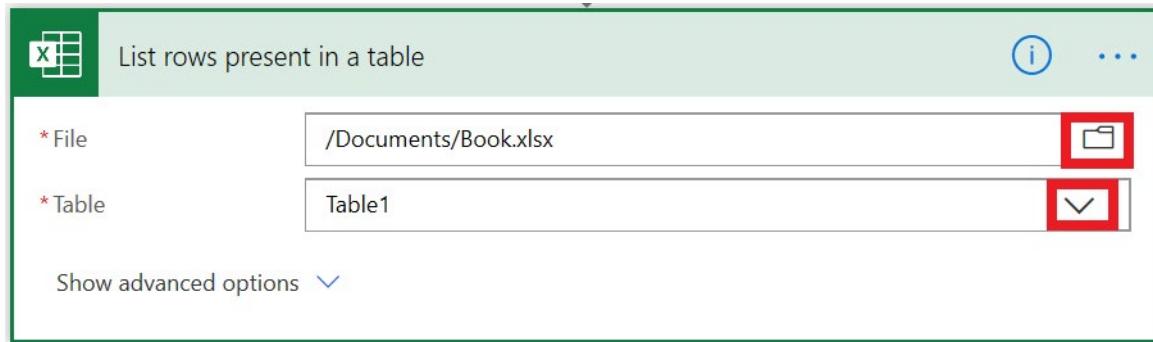
1. Sign in to **Power Automate**<sup>6</sup> by using your organizational account.
2. Select **My flows**.
3. Select **New**, and then select **Scheduled-from blank**.
4. Name your flow and under **Run this flow** set the flow to repeat every one Day.
5. Select **Create**.



6. Select **New step**, to add an action.

<sup>6</sup> <https://ms.flow.microsoft.com>

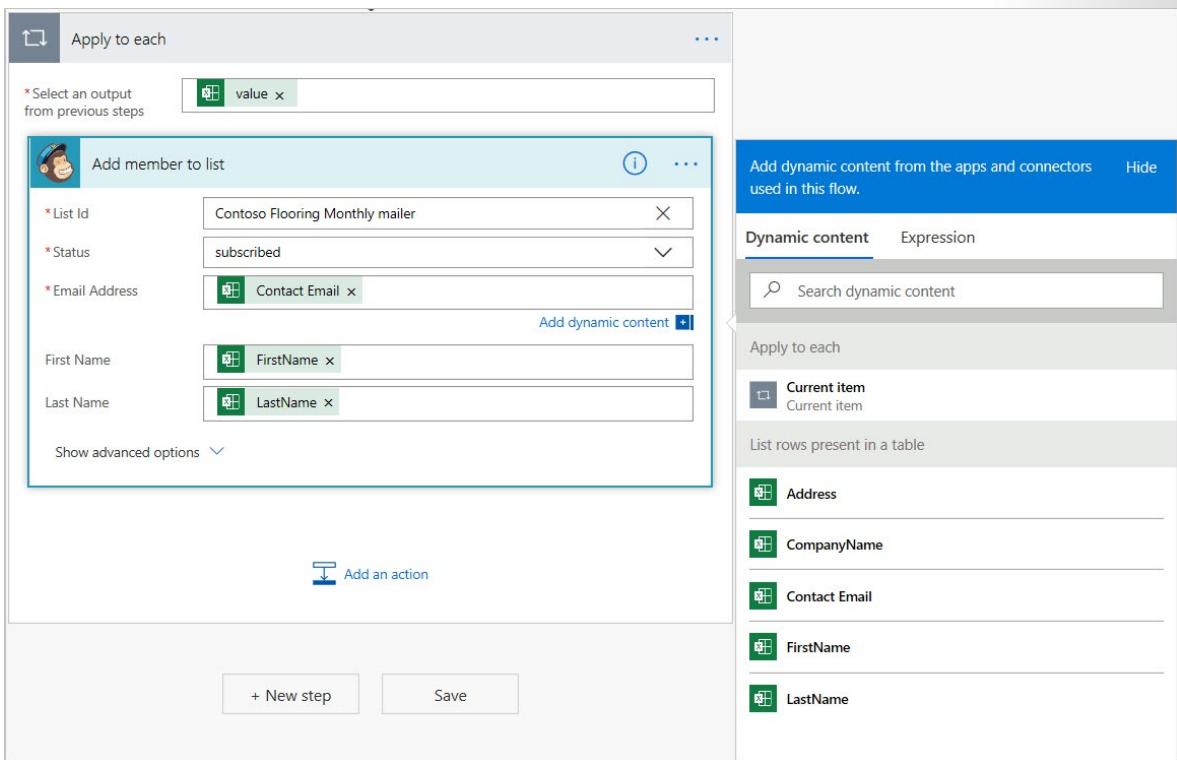
7. In the search field, enter *excel*, select the **Excel Online (Business)** service, and then select the **List rows present in a table** action.
8. In the **File name** field, select the folder button, and then select the Excel file to use.
9. In the **Table name** box, select the drop-down arrow, and then browse to and select the worksheet to use.



10. Select **New step**, and then select **Add an action**.
11. In the search field, enter *chimp*, select the **MailChimp** service, and then select the **MailChimp - Add member to list** action.

Note: MailChimp is a premium connector. Depending on your Power Automate license, you might need to sign up for a trial to use this connector.
12. In the **List Id** column, select the desired MailChimp mailing list. In the **Status** column, select *subscribed*.
13. In the **Email Address** field, use the dynamic content feature to add the **ContactEmail** field.

Notice that the flow automatically creates an additional step. Flow detects that you're setting up an action that requires an additional action. Whenever the flow reads a new email address, it will also create a new action for each row.
14. Use the dynamic content feature to fill in the **First name** and **Last name** fields.



And there you have it!

This flow will now run once a day, get the new rows from the Excel worksheet, grab the email address and name from each row, and enter the email address and name in the Contoso MailChimp mail list, saving you both time and money.

## Exercise - Send an email when a tweet is posted

You can create a flow that automatically performs one or more actions after it's triggered by an event. For example, the flow can notify you by email when someone posts a tweet that includes a keyword that you specify. In this example, posting a tweet is the event (also known as a *trigger*), and sending an email notification is the action. In this unit, you'll learn how to create this example flow.

### Prerequisites

- An account on [flow.microsoft.com](https://flow.microsoft.com)<sup>7</sup>
- A Twitter account
- Microsoft Office 365

### Specify an event to start the flow

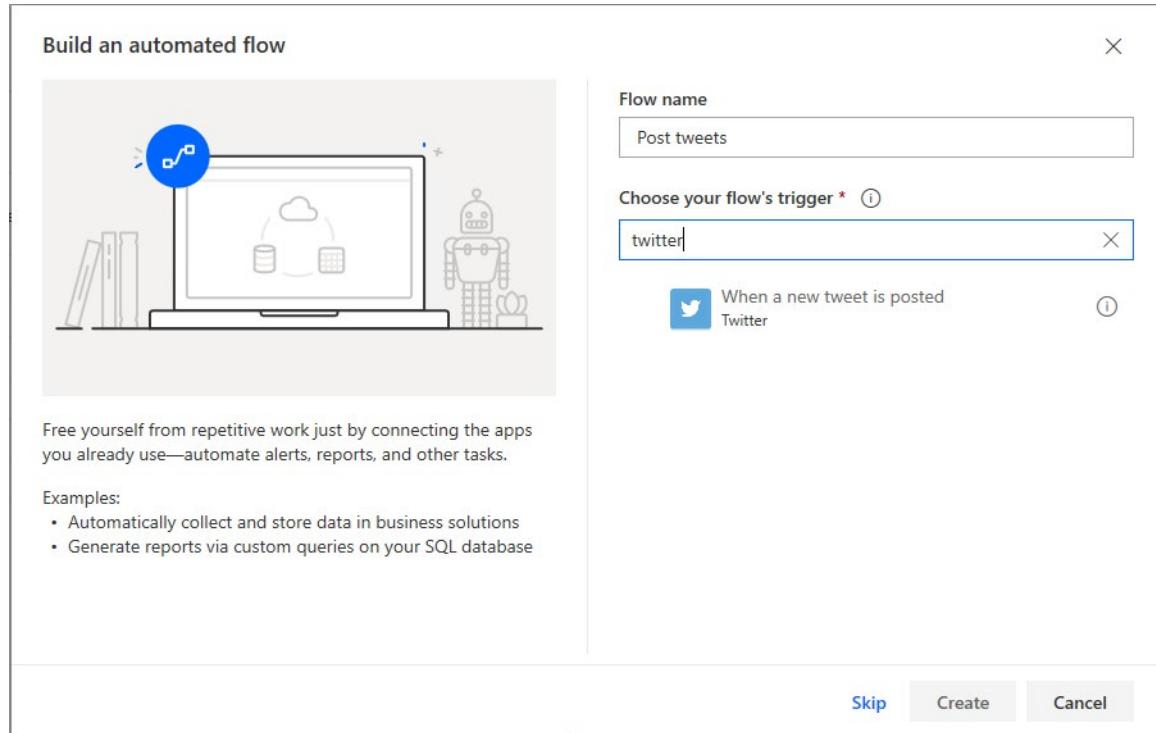
First, you must select the trigger (event) that starts the flow.

1. Sign in to **Power Automate**<sup>8</sup> by using your organizational account.

<sup>7</sup> <https://flow.microsoft.com>

<sup>8</sup> <https://flow.microsoft.com>

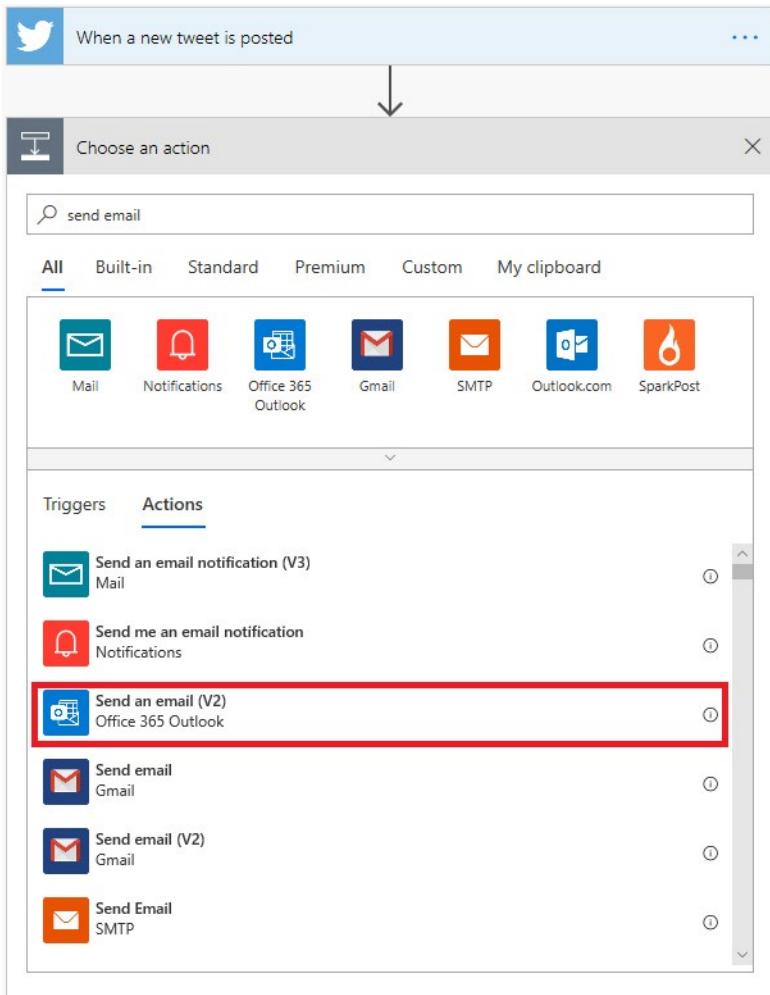
2. Select **My flows**.
3. Select **New**, and then select **Automated—from blank**.
4. Under **Choose your flow's trigger**, enter *twitter*, select the **Twitter - When a new tweet is posted** trigger and select **Create**.



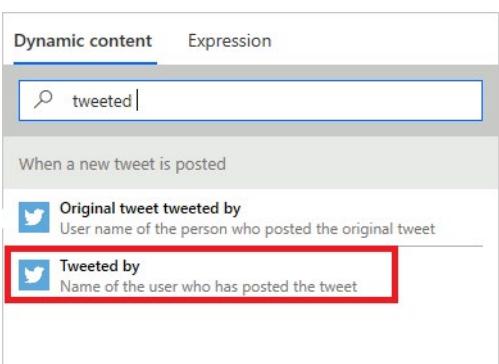
5. If you haven't already connected your Twitter account to Power Automate, select **Sign in to Twitter**, and then enter your credentials.
6. In the **Search text** box, enter the keyword to find.

## Specify an action

1. Select **New step**, and in the search column, enter *send email*, and then select the **Office 365 Outlook - Send an email** action.



2. If you're prompted to sign in, select the sign-in button, and then enter your credentials.
3. In the **To** field, enter or paste your email address, and then select your name in the list of contacts that appears.
4. In the **Subject** field, enter **New tweet from:** followed by a space.
5. In the list of dynamic content, select the **Tweeted by** token to add a placeholder for it.



6. Select the **Body** field, and then, in the list of dynamic content, select the **Tweet text** token to add a placeholder for it.

7. Optional: Add more tokens, other content, or both to the body of the email.
8. Select **Save** to save the flow.
9. Post a tweet that includes the keyword that you specified, or wait for someone else to post such a tweet.

Within a minute after the tweet is posted, an email message will notify you of the new tweet.

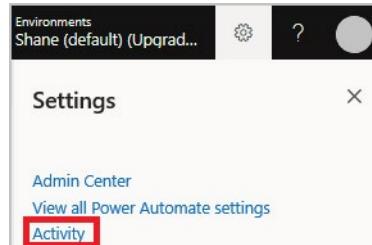
## Troubleshoot flows

In this unit, you'll learn how to troubleshoot common issues that might occur while you run your flows.

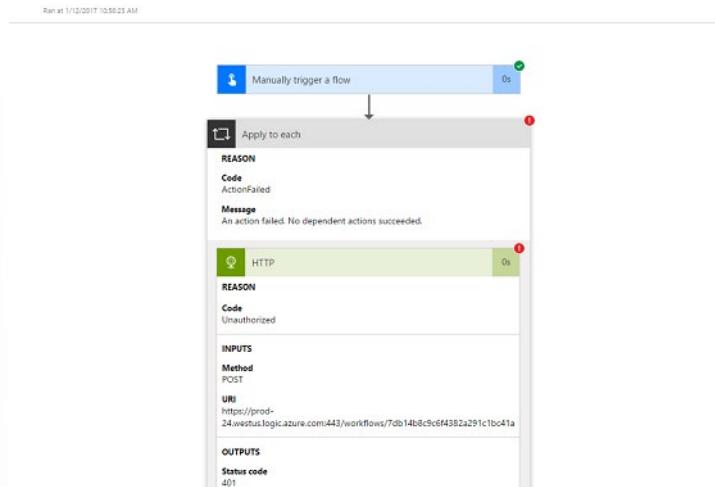
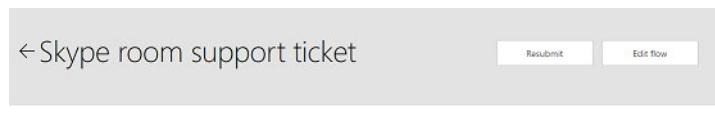
### Identify the error

Before you can fix a flow, you must identify why it failed. You will get an email with a list of failures each week.

1. Select the **Settings** button (the gear symbol) at the top of the web portal and select **Activity** (or select the **Activity** tab in the mobile app), and then select your flow in the list that appears.



2. Details about the flow appear, and at least one step has a red exclamation point (!) symbol. Open that step, and review the error message.



## Authentication failures

In many cases, flows fail because of an authentication error. If this type of error occurs, the error message includes the word "Unauthorized," or an error code of 401 or 403 appears. You can usually fix authentication errors by updating the connection.

1. You can view the connections by opening up the flow details by selecting the flow from **My Flows**.
2. Scroll to the connection that you saw the "Unauthorized" error message for.
3. Next to the connection, select the **Verify password** link in the message that states that the connection hasn't been authenticated.
4. Verify your credentials by following the instructions that appear. Then return to your flow-run failure, and select **Resubmit**.

The flow should now run as expected.

## Action configuration issues

Flows sometimes fail if a setting in one of the flow's actions doesn't work as expected. In this case, the error message includes the phrase "Bad request" or "Not found," or an error code 400 or 404 appears.

The error message should indicate how to fix the failure.

1. Select the **Edit** button, and then fix the issues inside the flow definition.
2. Save the updated flow, and then select **Resubmit** to try to run the flow again with the updated configuration.

## Temporary issues

If error code 500 or 502 appears, the failure is temporary or transient.

- Select **Resubmit** to try to run the flow again.

## Issues with your pricing plan

Sometimes your flows might behave unexpectedly because you aren't using the correct plan.

- To view your plan, in Power Automate, select **Learn**. It will redirect you to another page. Here select **Learn**, and then select **Pricing**.

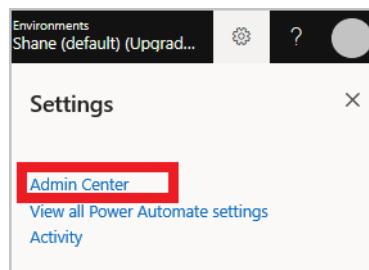
Learn more about **pricing and how to switch plans**<sup>9</sup>.

## Issues with data usage

You might have run out of data that you can use.

- If you're on a free plan or a trial plan, select the **Settings** button (the gear symbol) to show your current usage against your plan.

<sup>9</sup> <https://flow.microsoft.com/pricing/>



- If you're on a paid plan, runs are pooled across all users in your organization. We're working on features that will show information about available quotas and usage across an organization.

Important: If you exceed your data limit, Power Automate throttles your flow runs.

Learn more about [usage limits<sup>10</sup>](#).

## You might be running flows too often

Your plan determines how often your flows run. For example, your flows might run every 15 minutes if you're on the free plan. If a flow is triggered less than 15 minutes after its last run, it's queued until 15 minutes have passed.

Whenever a flow is triggered, whether by an automatic trigger or because you manually start it, the action counts as a run. Checks for new data don't count as runs.

Learn more about [usage limits<sup>11</sup>](#).

## You might be using an incorrect account

If you sign in by using a Microsoft account (for example, an account that ends with @outlook.com or @gmail.com), you can use only the free plan. To take advantage of the features of the paid plan, sign in by using your organizational account or school email address.

To upgrade, use an organizational account or a school account, or create a [Microsoft Office 365 trial account<sup>12</sup>](#).

## Some flows run more often than expected

Some flows might run more often than you expect. For example, you create a flow that sends you a push notification whenever your manager sends you an email. That flow must run every time you get an email from anyone, because the flow must check whether the email came from your manager. This action counts as a run.

## Other issues that are based on limits, and caveats

You might have issues that are based on other limits:

- Each account can have up to:
  - 250 flows.

<sup>10</sup> <https://flow.microsoft.com/pricing/>

<sup>11</sup> <https://flow.microsoft.com/pricing/>

<sup>12</sup> <https://powerbi.microsoft.com/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial/>

- 15 custom connectors.
- 20 connections per application programming interface (API) and 100 connections total.
- You can install a gateway only in the default environment.
- Some external connectors, like Twitter, implement connection throttling to control the quality of service. Your flows might fail when throttling is in effect. If your flows are failing, review the details of the run that failed in the flow's run history.

## Summary

Let's quickly review what we covered in this module.

In this module, you learned the basics of Power Automate, including the difference between triggers and actions and how to create flows for yourself or your team. You created a flow that automatically saves email attachments, one that alerts you of relevant tweets and a button flow to send yourself a reminder.

## Next steps

**Congratulations!** You've finished the first module of this learning path for Power Automate. Continue learning by checking out the next module in this learning path.

See you in the next module!

# Use the Admin center to manage environments and data policies in Power Automate

## Administer flows

Welcome to the Admin center! The Admin center is the central location where tenant admins and environment admins manage an organization's data policies and environments. Any changes that you make in the Admin center are immediately available to users in the organization.

In this module, you'll learn how to:

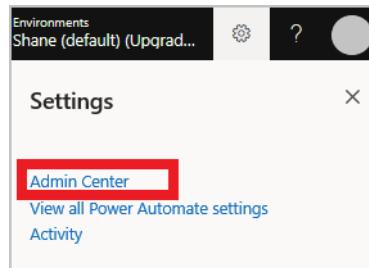
- Import and export flows.
- Share flows.
- Monitor flows.

## Open the Admin center

There are two ways to open the Admin center.

### Option 1: Power Automate settings

1. Go to **Power Automate**<sup>13</sup>, and sign in by using your organizational account.
2. Select the **Settings** button (the gear symbol), and then select **Admin Center** on the menu.



The Admin center is opened.

### Option 2: Open the admin center directly

- Go directly to the **Admin center**<sup>14</sup>, and sign in by using your work account.

## Environments

An *environment* is a space where you can store, manage, and share your organization's business data, apps, and flows. It also serves as a container to separate apps that might have different roles, security requirements, or target audiences.

<sup>13</sup> <https://flow.microsoft.com>

<sup>14</sup> <https://admin.flow.microsoft.com>

The way that you use environments depends on your organization and the apps that you're trying to build. Here are some examples:

- You can create separate environments to group the test and production versions of your apps.
- You can create separate environments that correspond to specific teams or departments in your company. Each environment holds the relevant data and apps for each team/department.
- You can create separate environments for different global branches of your company.
- You can build all your apps in a single environment.

## Data policies

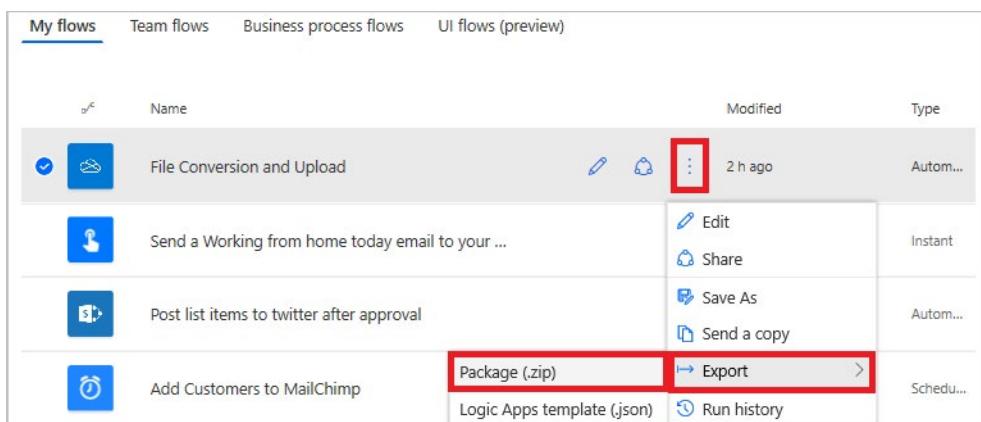
When you use Power Automate, the data is automatically protected with whatever security roles your organization already has in place. It isn't possible to use Power Automate to get access to information that the user doesn't already have access to in the organization. Some organizations may want to add an additional, optional layer of security that can proactively block flows that violate certain policies.

## Export and import flows

Power Automate allows you to export and then import a flow so that others can use it.

### Export a flow

1. Go to Power Automate, and sign in by using your organizational account.
2. In the left pane, select **My flows**.
3. For the flow that you want to export, select the **More commands** button (the three vertical dots), select **Export**, and then select **Package (.zip)**.



4. Fill in the package details:
  - **Name:** Enter a name for the flow.
  - **Environment:** Enter the environment for the flow.
  - **Description:** Enter a description of the flow.
  - **Review package content:** Select export options, and add comments to provide instruction or add version notes.

5. Select **Export** to export the zip file. You can then select the folder to in which to download.

When you export a flow, the dependent resources for your flow are also exported into the package.

## Import a flow

After a flow has been exported, anyone that you send the zip file to can import it.

1. Go to Power Automate, and sign in by using your organizational account.

2. In the left pane, select **My flows**.

3. Select **Import**.



4. On the **Import package** page, select **Upload**, and then, in the dialog box, select the zip file that you exported.

5. Back on the **Import package** page, select **Import**.

- In the flow settings, you can select whether to create a new flow or update an existing one with the flow definition from the package.
- You must also select the connections that are required to set up the flow as part of the import process.
- The **Import** button should become available after you've configured all the required settings.

6. After the flow is completely imported, you will have a link to open it and see the flow in **My Flows**.

## Learn how to distribute button flows

In the Power Automate mobile app, you can share button flows with other users or groups in your organization. The users or groups with whom you share a button can then run it the same way they run their own buttons.

You can also share a link to buttons that another person shared with you.

You can stop sharing your buttons at any time.

The screenshots in this unit were taken on a Apple iOS device. If you're using an Android or Windows device, what you see might differ, but the functionality is the same.

## Prerequisites

To share buttons, you need:

- An account that has access to **Power Automate**<sup>15</sup>.
- A flow to share.
- A mobile device that has the Power Automate mobile app for **Android**<sup>16</sup>, **Apple iOS**<sup>17</sup>, or **Windows Phone**<sup>18</sup>.

<sup>15</sup> <https://flow.microsoft.com>

<sup>16</sup> <https://aka.ms/flowmobiledocsandroid/?azure-portal=true>

<sup>17</sup> <https://aka.ms/flowmobiledocsios/?azure-portal=true>

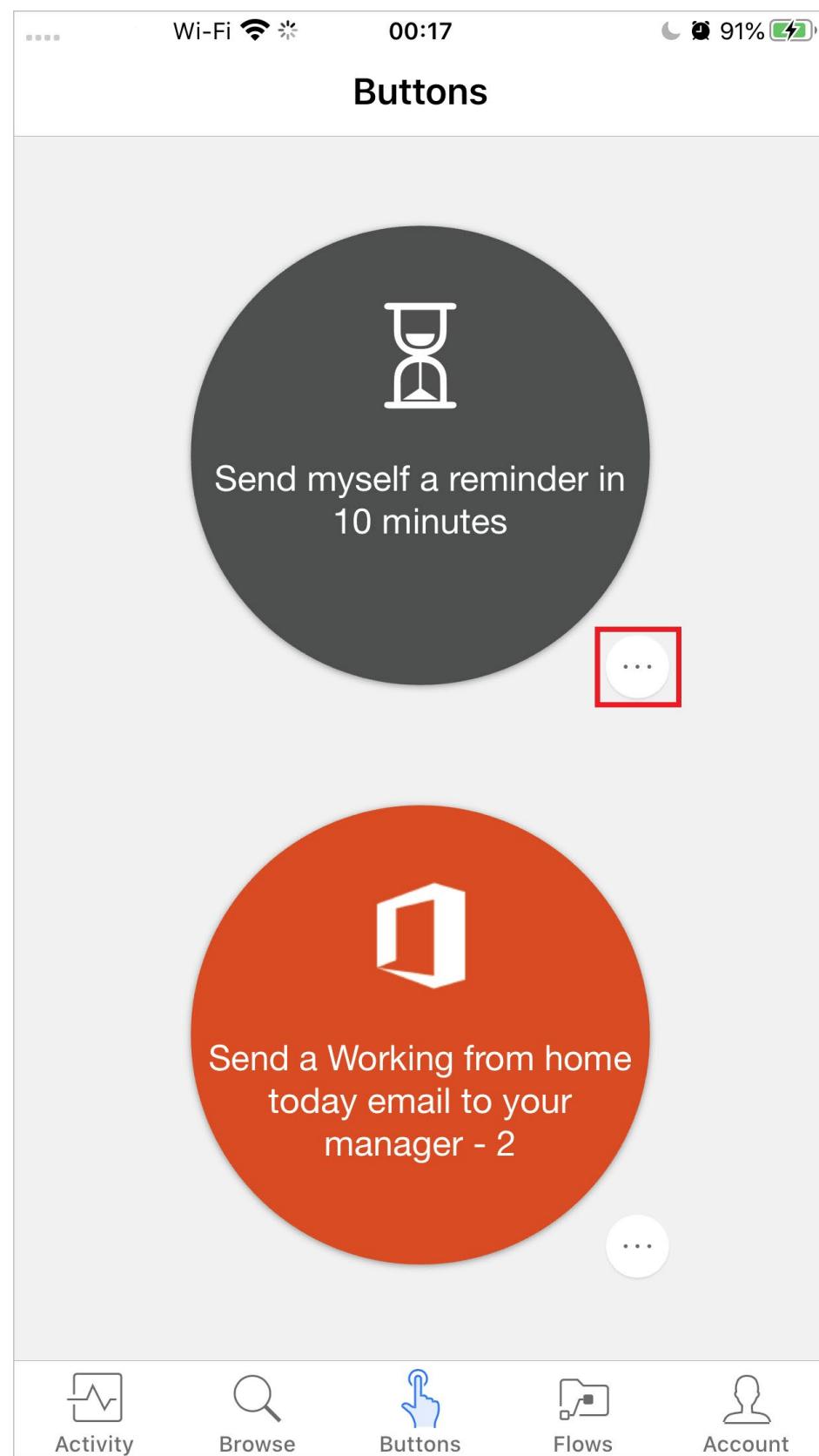
<sup>18</sup> <https://aka.ms/flowmobilewindows/?azure-portal=true>

- A person or group in your organization with whom to share your button.

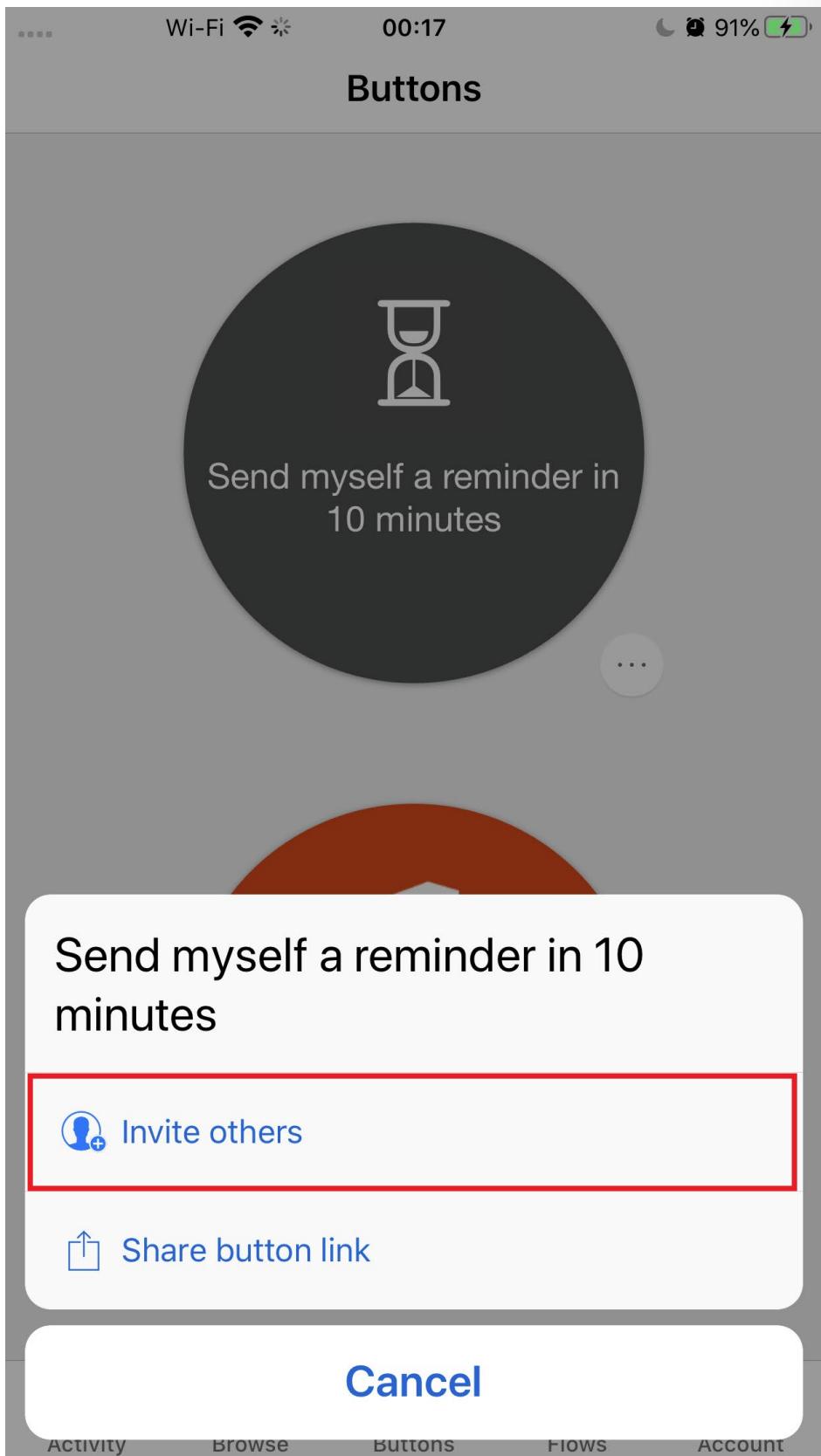
## Share a button

You share a button by using the **Buttons** tab of the Power Automate mobile app.

1. Start the Power Automate mobile app.
2. On the **Buttons** tab, select the three dots next to the button that you want to share.



3. On the pop up, select **Invite others**.



4. On the following page you can manage users with whom you have already shared the flow. For this exercise, select **Invite others** again to continue with sharing.
5. Search for and select the person or group that you want to share the button with.
6. Select **Send**.
7. On the page that states that the button sharing action was successful, select **Done**.

## Require users to use their own connections

When you share a button with other people, you can either let them use all the connections that the button uses or require that they use their own connections. Follow these steps to require the people with whom you share your button to use their own connections.

*Note:*

If you let other people use your connections, they can't access the credentials in your connection. They also can't reuse those connections in any other flow.

1. On the page that appears immediately after you share a button, select **Manage connections**.
2. Select **Edit** for the connection that you want to manage.
3. Select **Provided by user** or your email address to specify whose connections must be used in the shared button.

Wi-Fi ⚡ 00:35 96%

## Users and connections

Shared with Connections

Choose whether others will use your connection credentials or their own when running this button. [Learn more](#)

|                                                                                    |                                                 |      |
|------------------------------------------------------------------------------------|-------------------------------------------------|------|
|   | Notifications<br>Provided by run-only user      | Edit |
|   | Office 365 Outlook<br>Provided by run-only user | Edit |
|  | Office 365 Users<br>Provided by run-only user   | Edit |

Choose which connection to use for "Notifications"

@ .com

Provided by user

**Cancel**

Activity Browse Buttons Flows Account

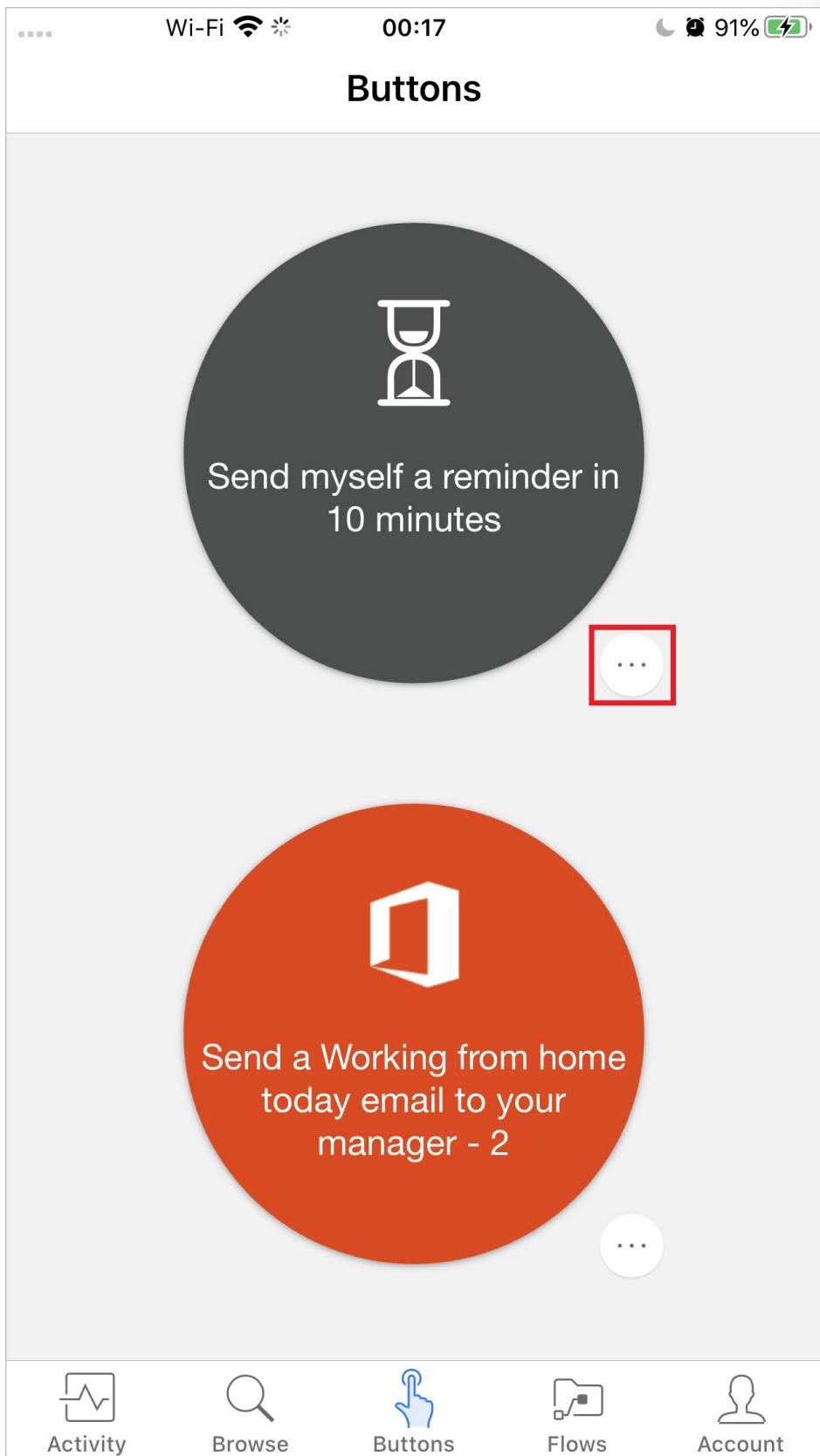
You can view or change your selection at any time.

1. On the **Flows** tab, select the flow that you shared.
2. On the **Users and connections** page, on the **Connections** tab, select **Edit** for the button that you want to manage.

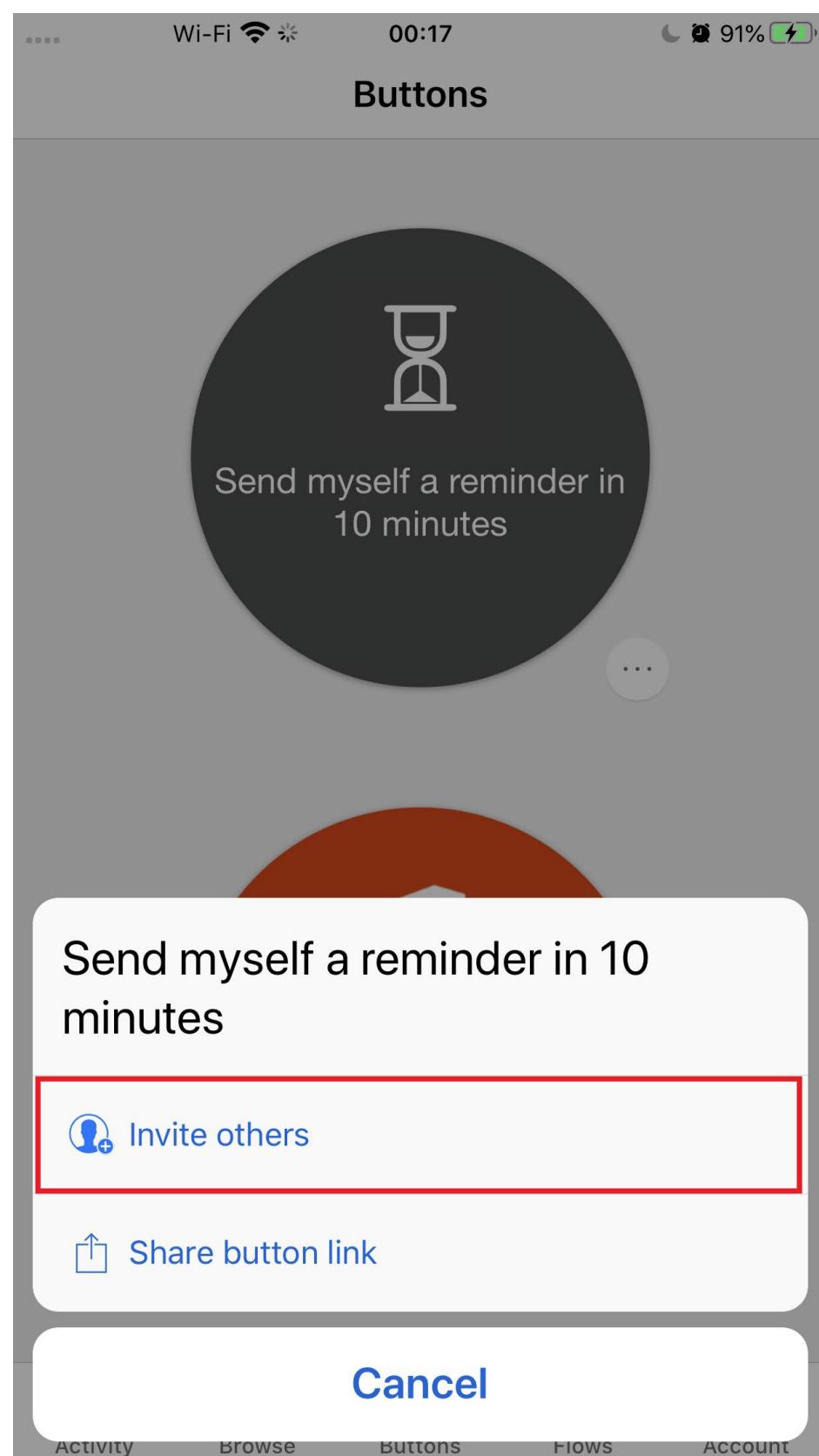
## Stop sharing a button

You can stop sharing a button by following these steps.

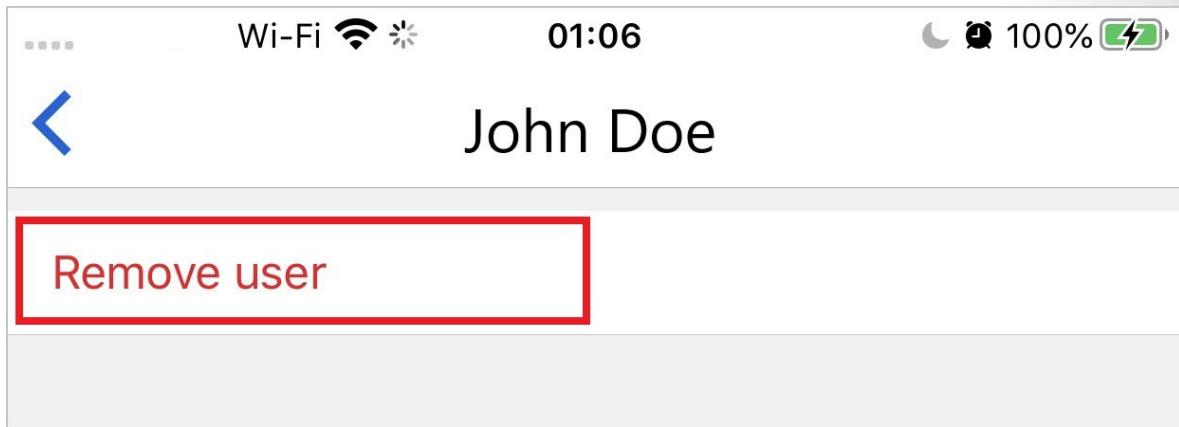
1. On the **Buttons** tab, select the three dots next to the button that you want to stop sharing.



2. On the pop up, select **Invite others**.



3. On the following page you can manage users with whom you have already shared the flow. On this page, select the user which you wish to remove.
4. On the user's page, select **Remove user**.



5. Wait for the removal action to be finished. The list on the **Button users** page is refreshed, and the user or group that you removed is no longer listed.

## Monitor the run history

You can view the whole run history, even for runs started by a person that a button is shared with.

1. Start the Power Automate mobile app.
2. Select the **Activity** tab to view the run history.

## Use shared buttons

Before you can run a button that someone has shared with you, you must add it to your **Buttons** tab from the **Add buttons** page.

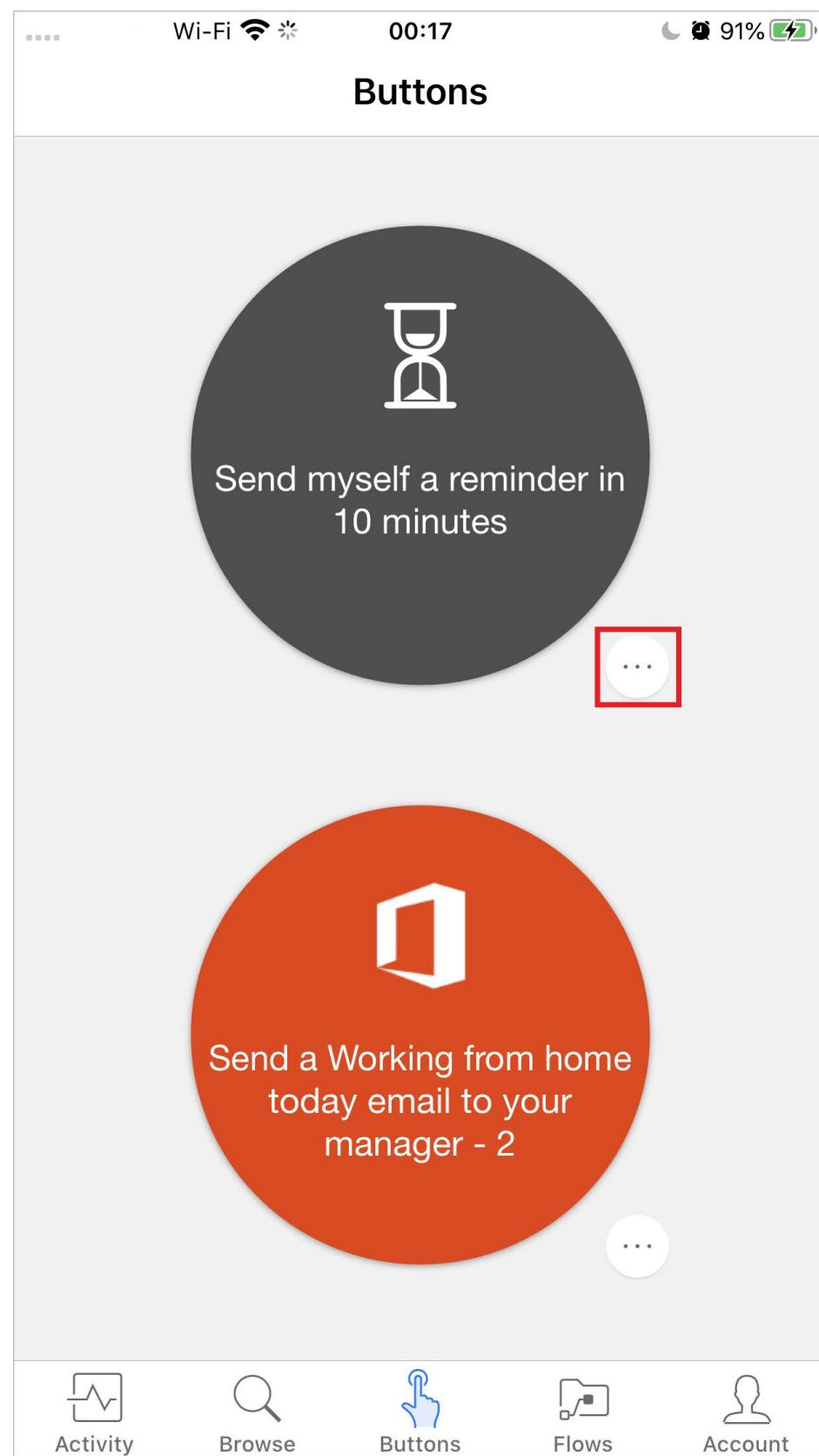
1. On the **Buttons** tab, select **Get more** (or the **New buttons are available** banner if it appears).
2. Select the button to use.

The button is immediately added to the **Buttons** tab. You can then run the button from the **Buttons** tab, just like any other button that's listed there.

## Re-share a button

You can share a link to a button that has been shared with you.

1. On the **Buttons** tab, select the three dots next to the button that you want to share.



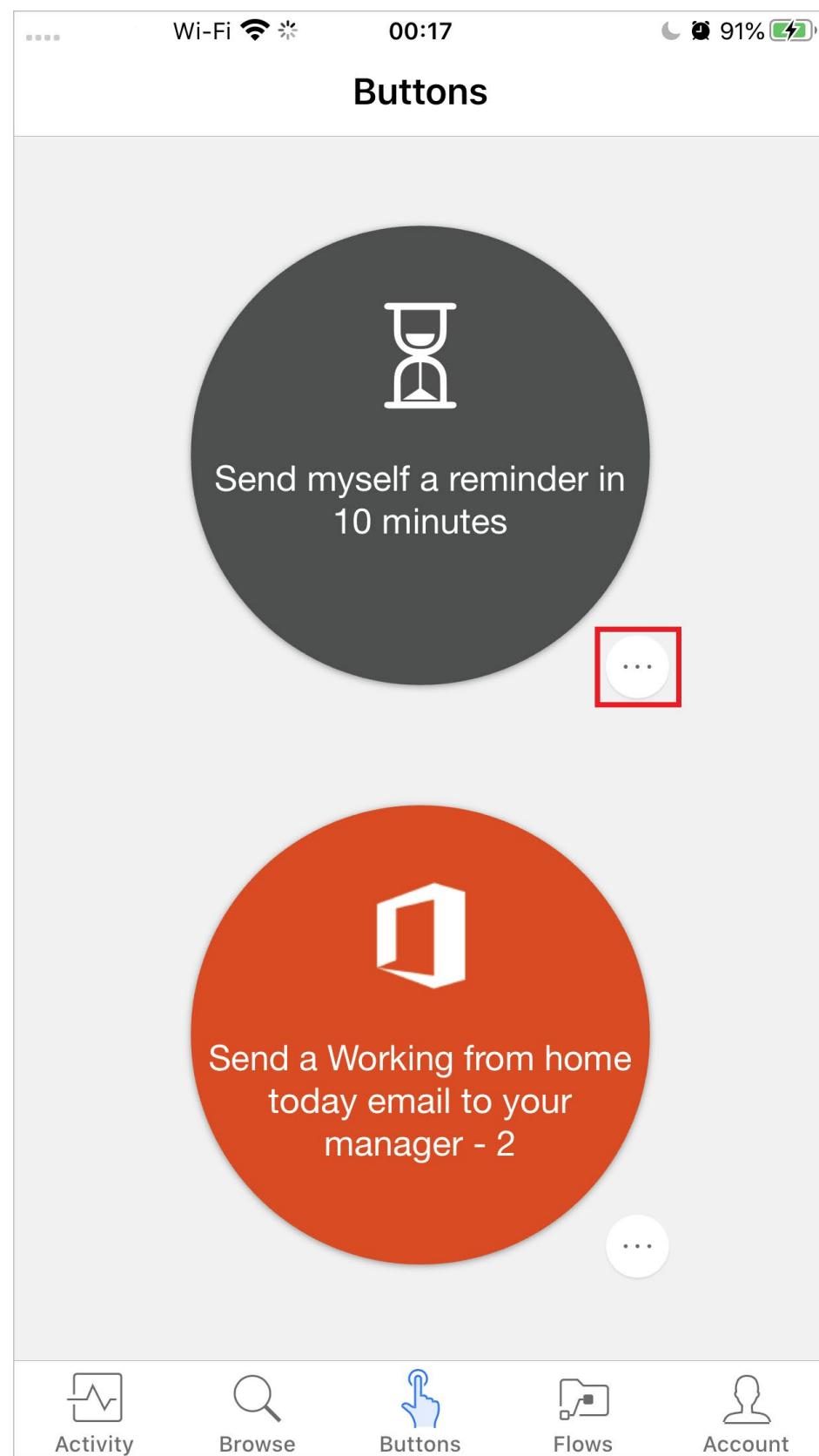
2. Select **Share button link**.

3. Select the app that you want to use to share the button.
4. Follow the steps for sending a button to a person that you want to share it with.

## Stop using a shared button

If you no longer want to use a button that was shared with you, you can remove it.

1. On the **Buttons** tab, select the three dots next to the button that you no longer want to use.



2. Select **Remove**.

*Note:*

After you remove a shared button, you can add it back by selecting **Get more** on the **Buttons** tab.

## Summary

Congratulations! You've grown your skills by administering flows.

In the previous modules of this learning path for Power Automate, you learned how to build simple and more complex flows.



Now you've learned how to take care of them when things aren't quite right. You learned how to:

- Export and import flows.
- Share button flows with other team members.
- Monitor flow activity.

## Continue your journey

Want to learn more about the Power Platform and how to solve problems? Check out these resources:

- **Power Apps<sup>19</sup>** learning path. You'll learn how to build apps that use flows to track or update business processes on any device.
- **Power BI<sup>20</sup>** learning path. You'll learn how to turn your unrelated sources of data into coherent, visually immersive, and interactive insights.
- **Customer Service<sup>21</sup>** learning path. You'll learn how to capture, track, and follow up on sales leads and connect to your CRM platform.

Even more exciting, you can do all of this without writing a line of code!

<sup>19</sup> <https://docs.microsoft.com/learn/paths/create-powerapps/>

<sup>20</sup> <https://docs.microsoft.com/learn/modules/get-started-with-power-bi/>

<sup>21</sup> <https://docs.microsoft.com/learn/modules/get-started-with-dynamics-365-for-customer-service/index>

# Introduction to business process flows in Power Automate

## Introduction to business process flows

A business process flow is a series of ordered work steps that a user completes within a business process. In Microsoft Power Automate, a business process flow is composed of a series of discrete stages that leads a user along a path toward process completion. Each stage contains one or more columns, called data steps, that you should complete before proceeding to the next stage in the business process flow.

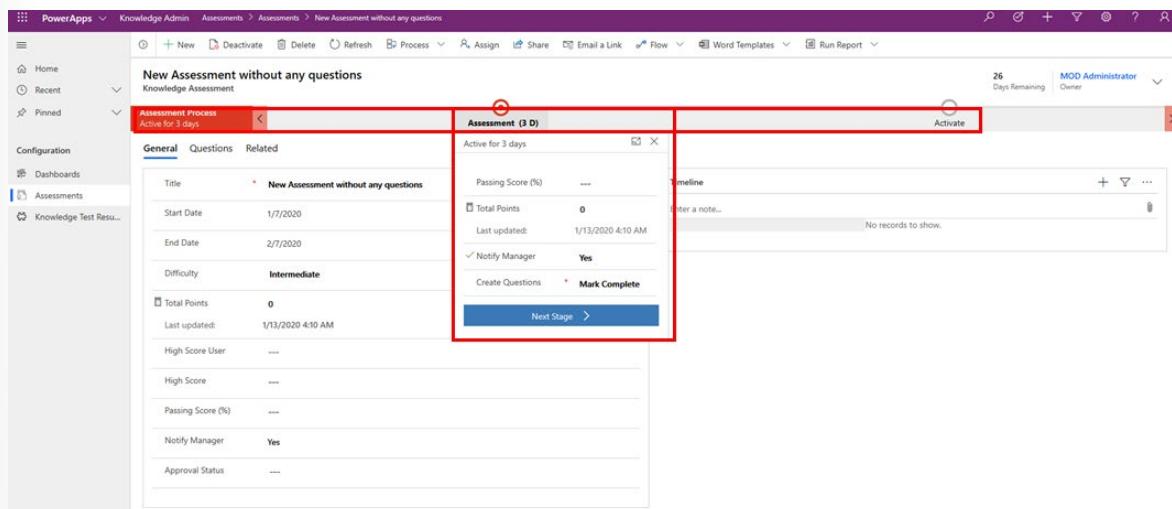
A business process flow visually guides a user through stages within a process and shows progress toward process completion. A user can also see which stages that they have completed and which stages that they still need to complete within an instance of a process.

Business process flows can be configured to require users to enter certain columns, data steps, before completing the stage. If needed, you can also allow users to jump stages. All data collected while you are completing a business process flow is stored in one or more tables in Microsoft Dataverse.

### [!NOTE]

To complete this exercise, you will need access to an account that has permission to create tables and columns in Dataverse and you need a Power Apps or Dynamics 365 license. Ask your Power Platform administrator for proper permissions or sign in and create a personal Power Apps and Power Automate development environment by using the Power Apps Community Plan (which is free). Sign up by accessing the Power Apps Community Plan page.

The following illustration shows a simple business process flow with two stages for adding questions to a survey. The business process flow is shown as a component of a model-driven solution in Power Apps.



Business process flows are created and managed by using Power Automate, and they are available for out-of-the-box tables and for your own custom tables. Microsoft has many prebuilt business process flows, or you can create your own. Additionally, a table can have none, one, or many business process flows associated with it. Business process flows are customizable to fit many organizational needs.

### *Tip:*

Business process flows are meant to guide users through steps that are required to

complete a business process. You must have a valid Power Apps Dynamics 365 license or a valid Power Apps license to create or use a business process flow.

## Business process flow vs regular

A business process flow is a visual guide meant to help users complete a business process by using a set of predefined stages. Users are not limited on how long they run a business process or how long they have a stage open. All data associated with the business process flow can only be stored in one or more tables in Microsoft Dataverse. Business process flows can only have Dataverse as a data source. Dataverse has many different out-of-the-box business process flows that you can use as-is or modify them to fit your needs.

A Power Automate flow does not have any visual components like a business process flow. Power Automate flows can be configured to work with many different data sources, and a flow can connect to many different data sources within the same flow. A flow can be configured to time out if it is not completed in a certain time and can be triggered to move between steps based on data or user interaction. Flows support complex logic and looping, and a Power Automate flow can call another Power Automate flow as needed.

## Business process flows value to organizations

Business process flows allow organizations to quickly standardize how processes are completed and what data is collected at each stage. Business process flows support logical branching so they can be used to standardize many common business processes within an organization.

Business process flows offer the following benefits:

- Improved outcomes
- Consistent stages and work steps across all instances of the process
- Improved data collection and reporting
- Decreased time to complete the process
- Predictable outcomes

Business process flows are simple to set up and administer. Business process users who are close to business operations and processes can create new business process flows or modify out-of-the-box business process flows by using Power Automate.

Business process flows can be customized based on security roles, allowing access to the appropriate stages and steps based on a security role. Finally, the process of each instance of a business process flow can be monitored, and the data from the process flow can be used in Power BI dashboards and reporting for simplified administration.

## Business process flows and the larger Power Platform

Business process flows are deeply integrated with the Microsoft Power Platform. They are created, customized, and managed by using Power Automate. You can manage or create new business process flows by launching Power Automate and selecting **Business process flows** under **My Flows** in Power Automate, as shown in the following image.

The screenshot shows the Microsoft Power Automate interface. On the left is a navigation sidebar with options like Home, Action items, My flows (selected), Create, Templates, Connectors, Data, AI Builder, Solutions, and Learn. The main area is titled 'Flows' and has tabs for 'My flows', 'Team flows', 'Business process flows' (which is highlighted with a red box), and 'UI flows (preview)'. Below these tabs is a table with columns for 'Display name' and 'Last modified'. Three entries are listed: 'Translation Process' (modified 1 wk ago), 'New Process' (modified 1 wk ago), and 'Expired Process' (modified 1 wk ago).

You can launch the new business process flows as a component of a model-driven Power App or a stand-alone application within Power Automate (called an immersive business process flow), as shown in the following image.

The screenshot shows a Power App for a 'San Francisco Smog Check' with a 'Customer Check In' step. The flow continues through 'Customer Info', 'Automobile Info', '1971 Or Older Smog Check', and ends at 'Collect Payment'. The '1971 Or Older Smog Check' step is expanded, showing a table with rows for 'Smog Pump Intact' (Yes), 'PVC Valve Intact' (Yes), 'Aftermarket Exhaust Headers' (No), and a comment 'Comments Pre 1972' (Great car). The 'General' section on the left contains fields for Entry Date (1/17/2020), Invoice Number, Amount (\$300.000), Payment Type (Visa), Invoice Title (San Francisco Smog Check), Owner (Dylan), First Name (Dylan), Last Name (Hope), Location (Los Angeles), Smog Cert Number (45,450,009), and Service Comments (---).

As mentioned previously, all data that is associated with a business process flow is stored in one or more Dataverse tables (custom or standard). You could launch an instant Power Automate flow in conjunction with a business process flow to store data outside of Common Data Service if needed. Additionally, you can create Power BI dashboards from the data that was collected within the business process flow.

Business process flows are deeply integrated within Power Platform and offer powerful ways to improve how you manage common business processes.

## Uses of business process flows

Knowing when to use a business process flow or a regular Power Automate flow to automate a processes or task can be difficult. The following guidelines can help you decide if you should use a business process flow or a standard Power Automate flow.

Use a business process flow if you:

- Want to create automated business processes with Dynamics 365 Solutions.
- Want a simple visual guide to help users complete a process.
- Want to use out-of-the-box business process or custom business process flows.
- Have a Dynamics 365 license and want to create automated business processes with Dataverse.

Use a Power Automate flow if you:

- Want to schedule a workflow to start based on a predefined time interval or after X minutes, hours, or days of some action or event.
- Want to trigger a flow based on data outside of Dataverse (SharePoint, for example).
- Do not want to store data that is captured in the flow in Dataverse.
- Want to push notifications outside of Outlook (SMS or Gmail, for example).
- Want to use and create workflows with only an Office 365 license.

## Summary

Business process flows are a powerful way to visually standardize and guide users through a process. They ensure consistent implementation and data collection, can improve the quality of outcomes, and decrease the cycle times to complete a process.

You can create business process flows as a custom solution, or you can choose to use or customize out-of-the box business process flows that are shipped with Dataverse and Dynamics solutions.

Additionally, you can create business process flows as a stand-alone solution (called an immersive business process flow), or you can include them as a panel within a model-driven app.

An important parameter to remember is that business process flows only store data within Dataverse.

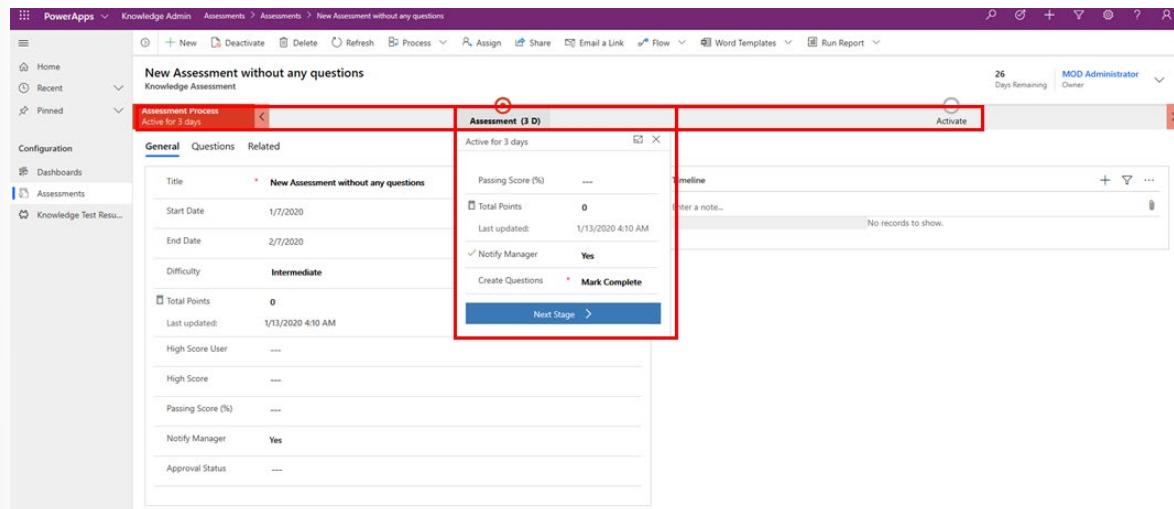
You should use a regular Power Automate flow if you want to store data in a data source outside of Dataverse, or if you want to trigger the flow based on a time or recurring condition or if you only have an Office 365 license available.

# Create an immersive business process flow in Power Automate

## Introduction to immersive business process flows

Business process flows are available in two different varieties: embedded within a model-driven app and as a stand-alone solution called an immersive business process flow. You should examine the differences between each before learning how to create an immersive business process flow.

Embedded business process flows are visual representations of a series of work steps within an end-to-end process. They always appear along the top of a model-driven app in their own control, as shown in the following image.



An immersive business process flow can exist on its own, and it is built entirely within the Power Automate editor. They will always exist within the default solution in an instance of Microsoft Dataverse, and are only accessible within Power Automate or the data tab that is associated with an entity in Dataverse. Immersive business process flows offer many of the same advantages of an embedded business process flow, but with the added advantage of simplified creation and streamlined management.

You will learn how to build an immersive business process flow in the next units of this module. The following screenshot shows a simple two-step immersive business process flow.

By taking a closer look, you'll see that immersive and embedded business process flows look similar. The map of the business process flow is shown along the top of the screen and a main form is shown beneath it.

The name of the business process flow is shown on the left side of the control, while each step (called a Stage) is shown as a red circle. A user selects the red circle of the current Stage and fills out information in a drop-down screen that includes various fields (called data steps).

The key difference between immersive and embedded business process flows is that an embedded business process flow is a component within the larger model-driven app, while the immersive business process flow is a stand-alone solution. Embedded and immersive business process flows help users complete a process within the context of a larger solution.

A user always views and completes an embedded business process flow within a model-driven app. An immersive business process flow is a stand-alone app, and it is built solely for completing the business process. The immersive business process flow exists as the entire solution rather than a component of a larger model-driven app.

Many of the concepts and techniques that are discussed and demonstrated within this learning path apply to both embedded and immersive business process flows. Immersive and embedded flows are similar. The main difference between them is how they are launched and the context that they are viewed in.

## Exercise - Create an immersive business process flow

The following scenario and exercise will help you practice building an immersive business process flow with Power Automate and Microsoft Dataverse. Remember, all data that is associated with any business process flow is always stored in one or more tables within Dataverse and business process flows.

### Note:

To complete this exercise, you will need access to an account that has permission to create tables and columns in Dataverse and you should have a Power Apps or Dynamics 365 license. Ask your Power Platform administrator for proper permissions or sign in and create a personal Power Apps and Power Automate development environment by using the Power Apps Community Plan (which is free). Sign up by accessing the [Power Apps Community Plan<sup>22</sup>](https://powerapps.microsoft.com/communityplan/?azure-portal=true) page.

<sup>22</sup> <https://powerapps.microsoft.com/communityplan/?azure-portal=true>

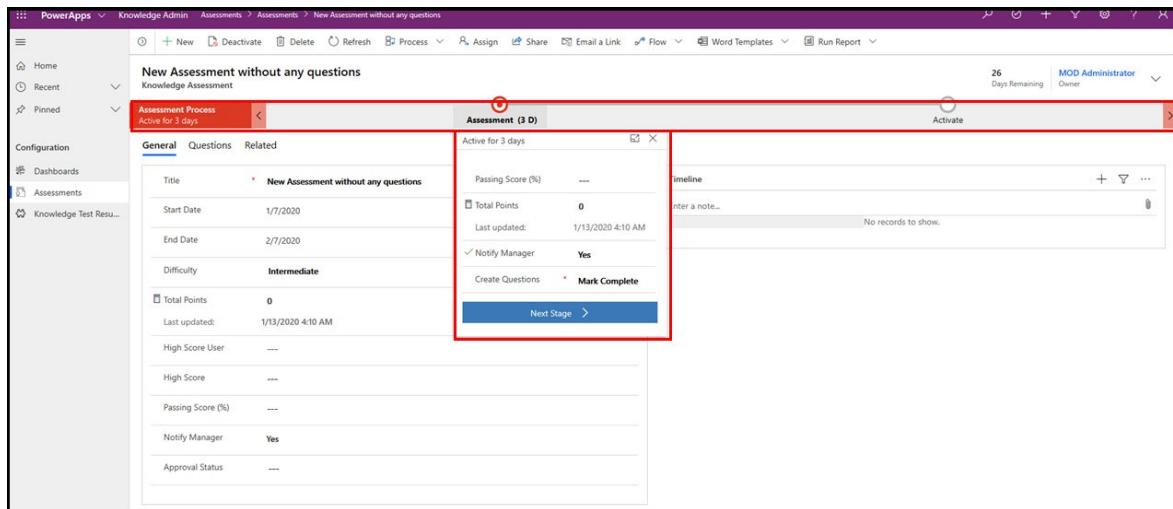
## Scenario

You work for SmogChecksRUs, a rapidly growing auto repair company that specializes in performing automotive smog checks and other auto services. SmogChecksRUs has been using a Microsoft Excel spreadsheet to collect customer and vehicle information, but now the marketing department has asked for a better way to collect information so they can follow up with customers and schedule checkups every two years, improve customer retention, and increase sales.

Management believes that improved data collection and standardized processes will improve customer satisfaction, improve customer loyalty, and increase recurring business and overall sales revenue. You have decided to create an immersive business flow by using Power Automate to meet management goals.

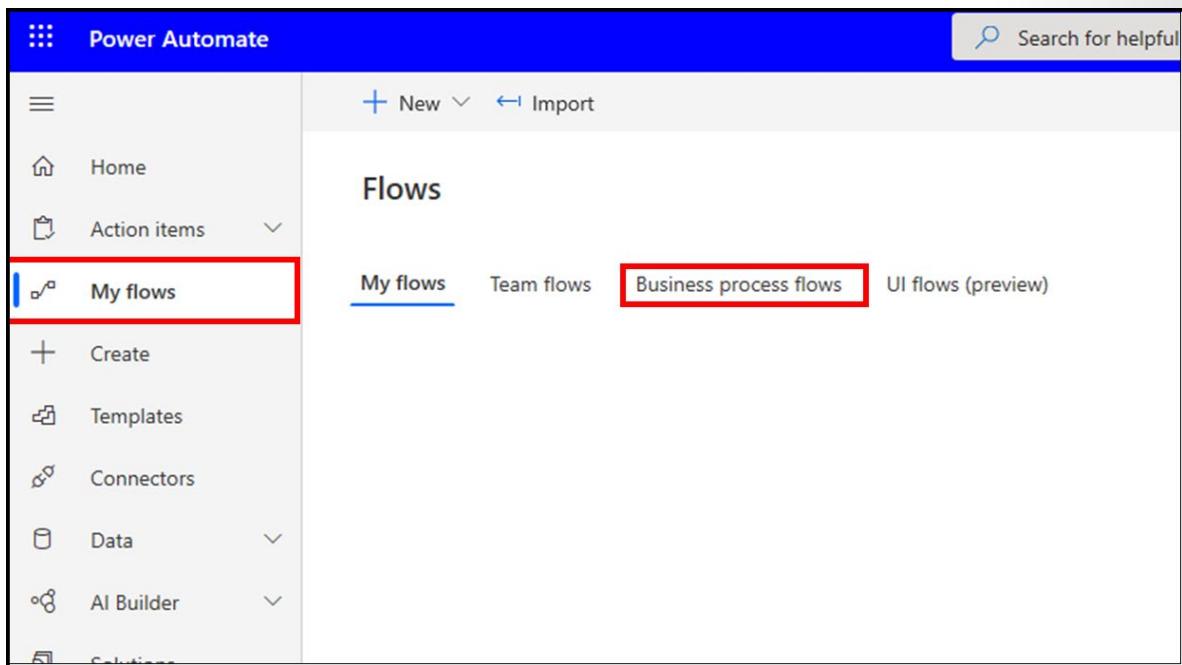
## Create an immersive business process flow

1. Go to **Power Automate**<sup>23</sup> and sign in to your local instance.
2. Select the proper environment in the upper-right corner of the screen.

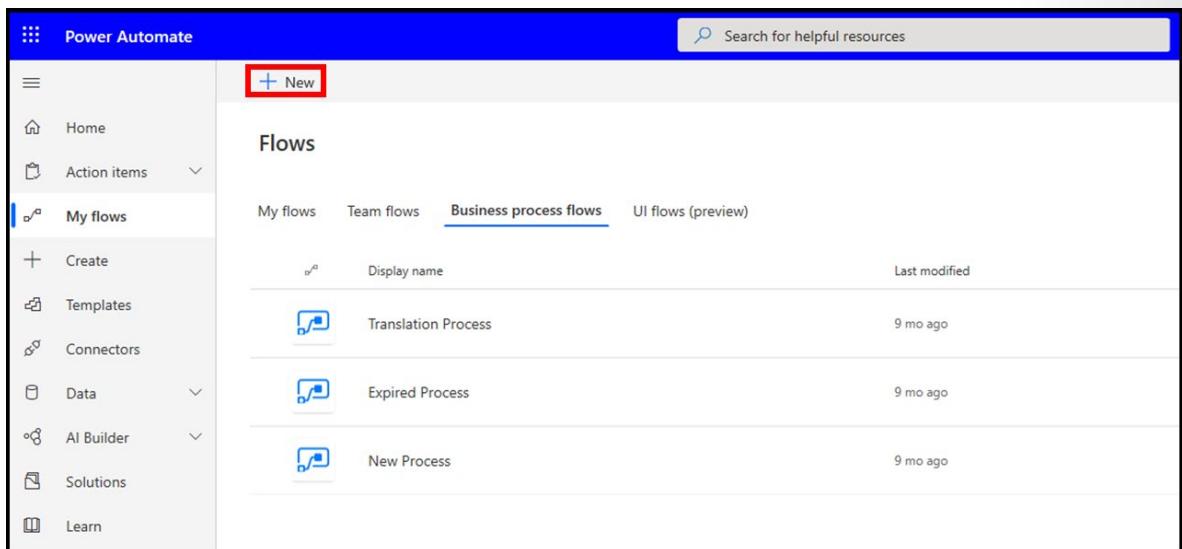


3. Select **My Flows** on the menu on the left side of the screen and then select the **Business process flows** tab.

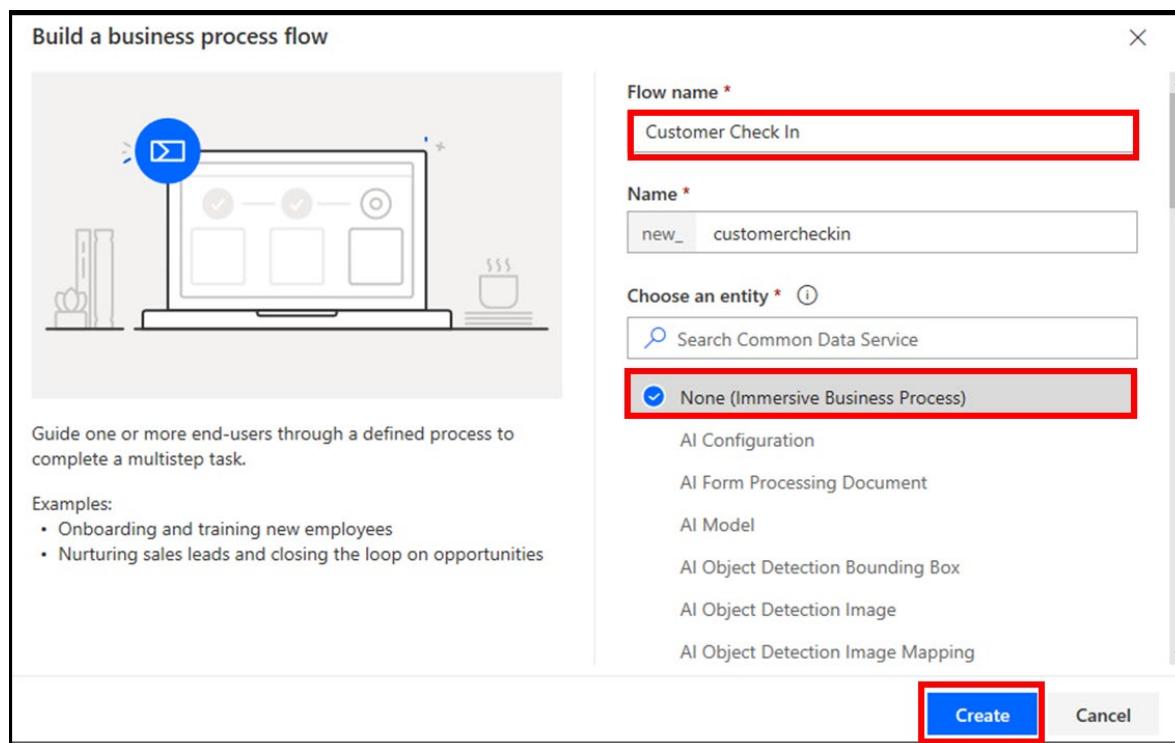
<sup>23</sup> <https://preview.flow.microsoft.com/?azure-portal=true>



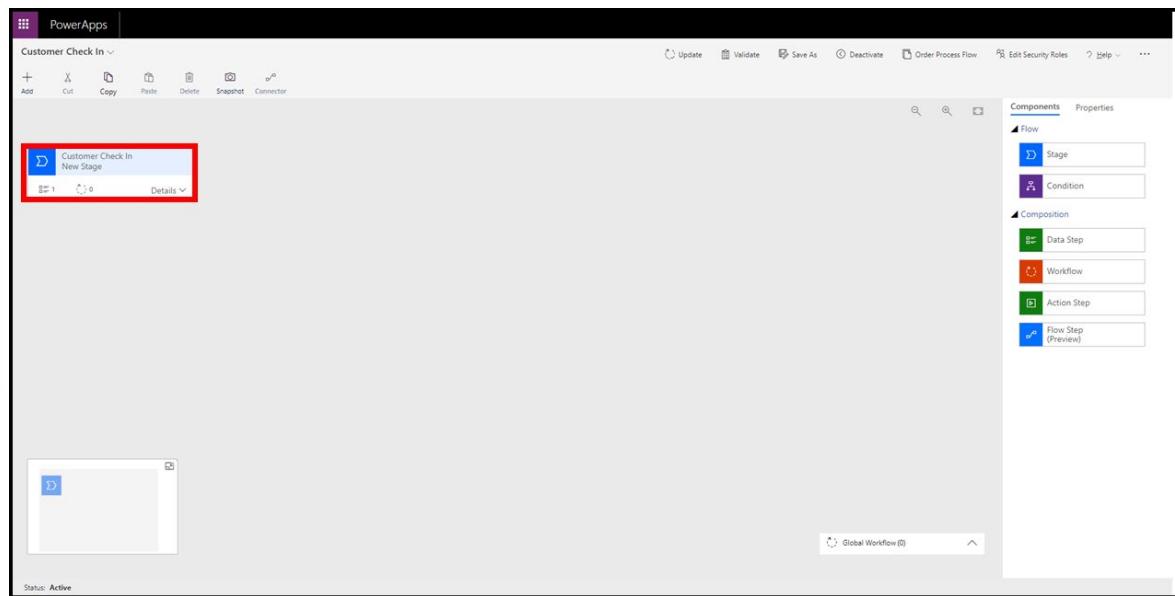
4. Select the **+ New** button on the upper-left corner of the screen.



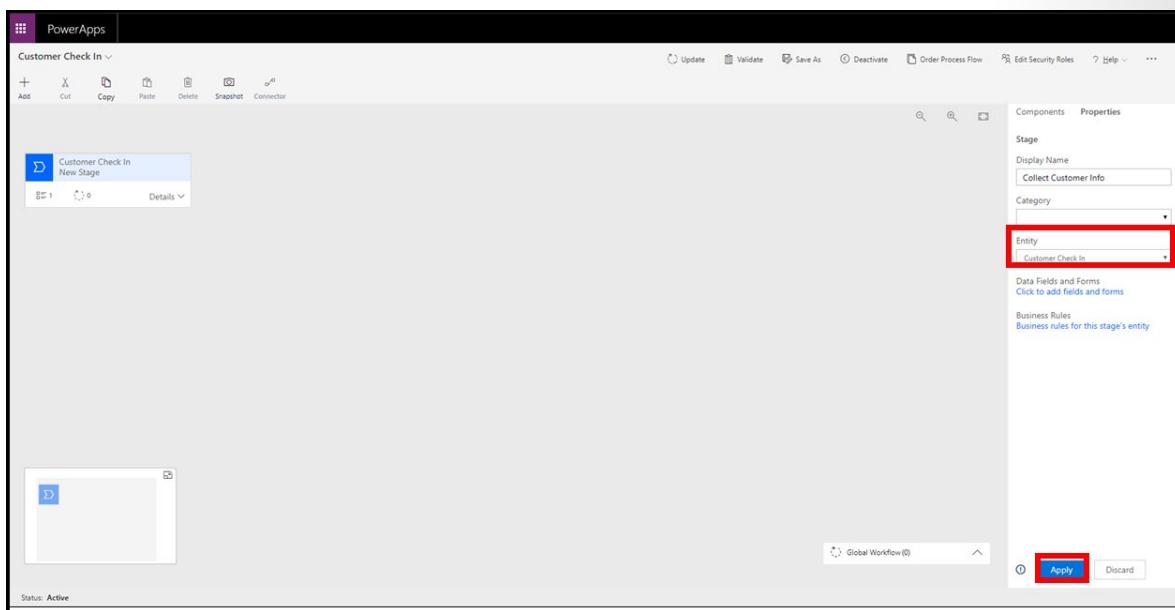
5. Enter **Customer Check In** as the **Flow name**, select **None (Immersive Business Process)** under the table drop-down list, and then select the **Create** button, as shown in the following screenshot. Wait a minute for the table to be created, and then the **Business process flow** editor will launch.



6. In the **Business process flow** editor, add columns to the new table (customercheckin) that was automatically created in the previous step. Select **Customer Check In New Stage**, as shown in the following figure.

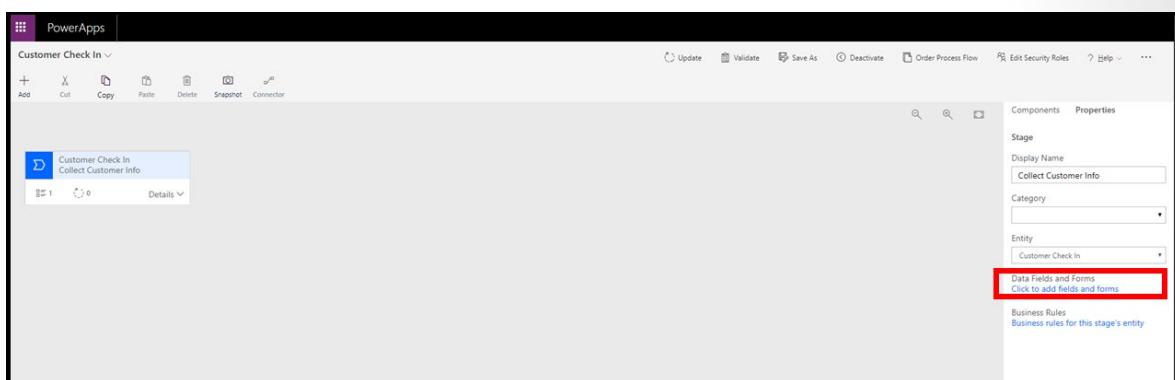


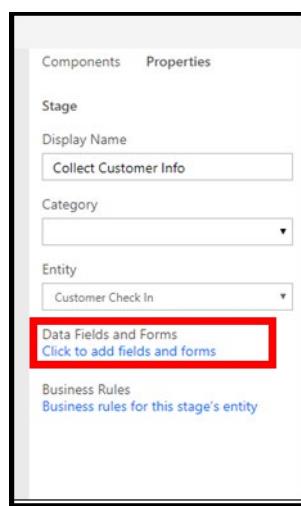
7. Select **New Stage** within the designer, rename the Stage to **Collect Customer Info**, and then select the **Apply** button, as shown in the following screenshot.



Next, you will add columns to the new **CollectCustomerInfo** table in Dataverse so you can capture customer information.

8. Select the **Collect Customer Info** stage and then select the **Click to add columns and forms** hyperlink, as shown in the following figure.





*Note:*

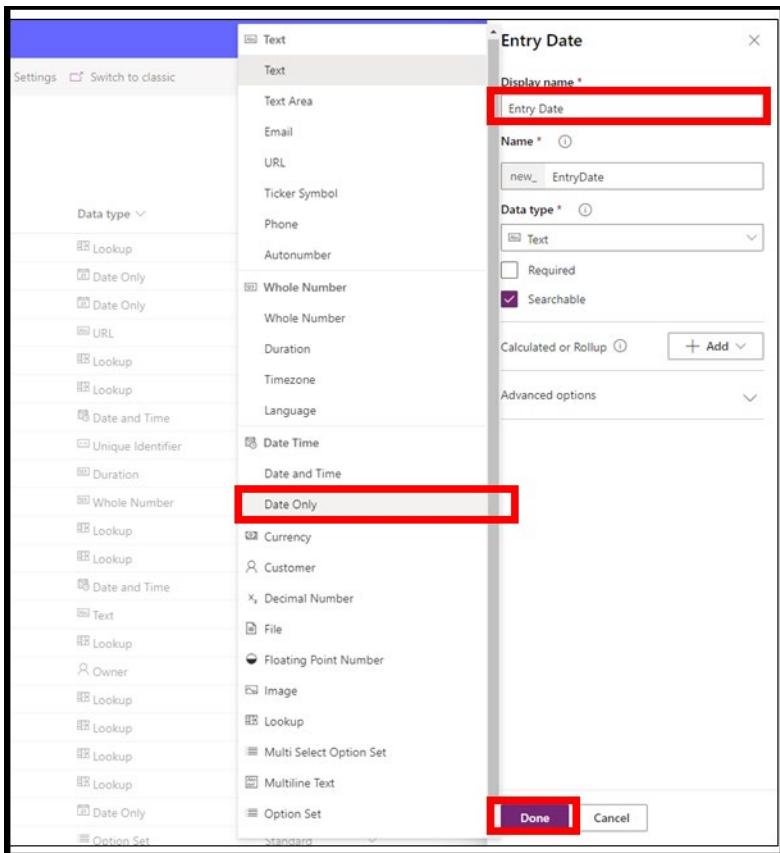
If you select **Click to add field and forms** link from Power Automate, you may see the classic UI. To see the modern UI, Open Power Apps in a new window and sign-in. On the left, select **Data** then **Entities**.

Find and select the **Customer Check In table**. Add the columns as listed in the next step.

9. Add columns to the collectcustomerinfo table by selecting the **Add column** button, as shown in the following screenshot.

| Display name ↑          | Name ↓               |
|-------------------------|----------------------|
| Active Stage            | activestageid        |
| Active Stage Started On | activestagestartedon |
| Completed On            | completedon          |

10. Add each of the columns from the following list. Enter the name and data type and then select the **Done** button each time you add a new column, as shown in the following screen.



- Entry Date - Date Only
- First Name - Text
- Last Name - Text
- Address - Text
- City - Text
- State - Text
- Postal Code - Text
- Phone Number - Phone
- Comments - Text Area

11. When you are finished, select the **Save Table** button to save the new columns. Make sure that you select the **Save Table** button or none of the columns will be added.

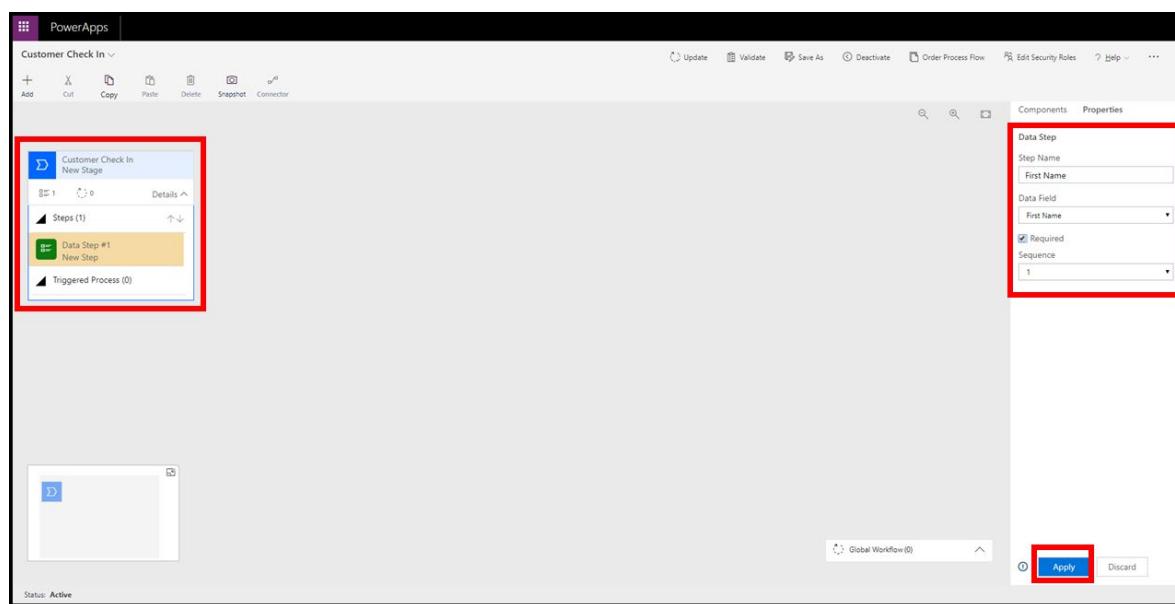
12. Close the current tab of the browser that is showing the table columns, and then return to the business process flow designer screen.

## Add columns as steps and finish the flow

Now, you will add the columns as Steps in the first Stage in the **Customer Check In** business process flow.

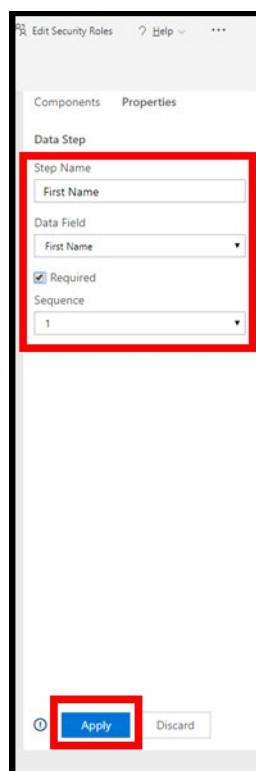
1. Refresh your browser and then select the **Customer Check In** Stage.

2. Select the **Details** down arrow to Expand

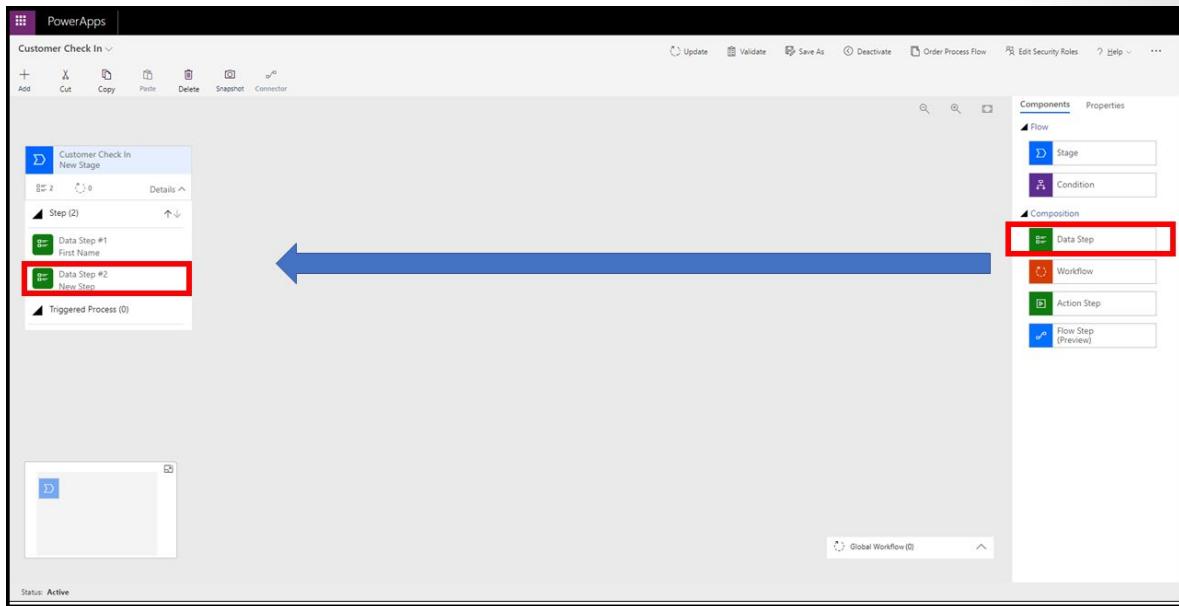


3. Select **Data Step # 1**, and then enter the following information:

- Step Name - Enter **First Name**.
- Data Field - Select **First Name** from the drop-down menu.
- Select the **Required** check box.
- Select the **Apply** button.

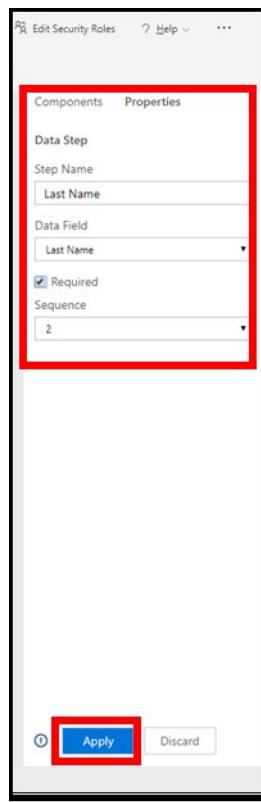


4. Select the **Components** tab and then drag a Data Step under **Data Step #1**, as shown in the following figure.

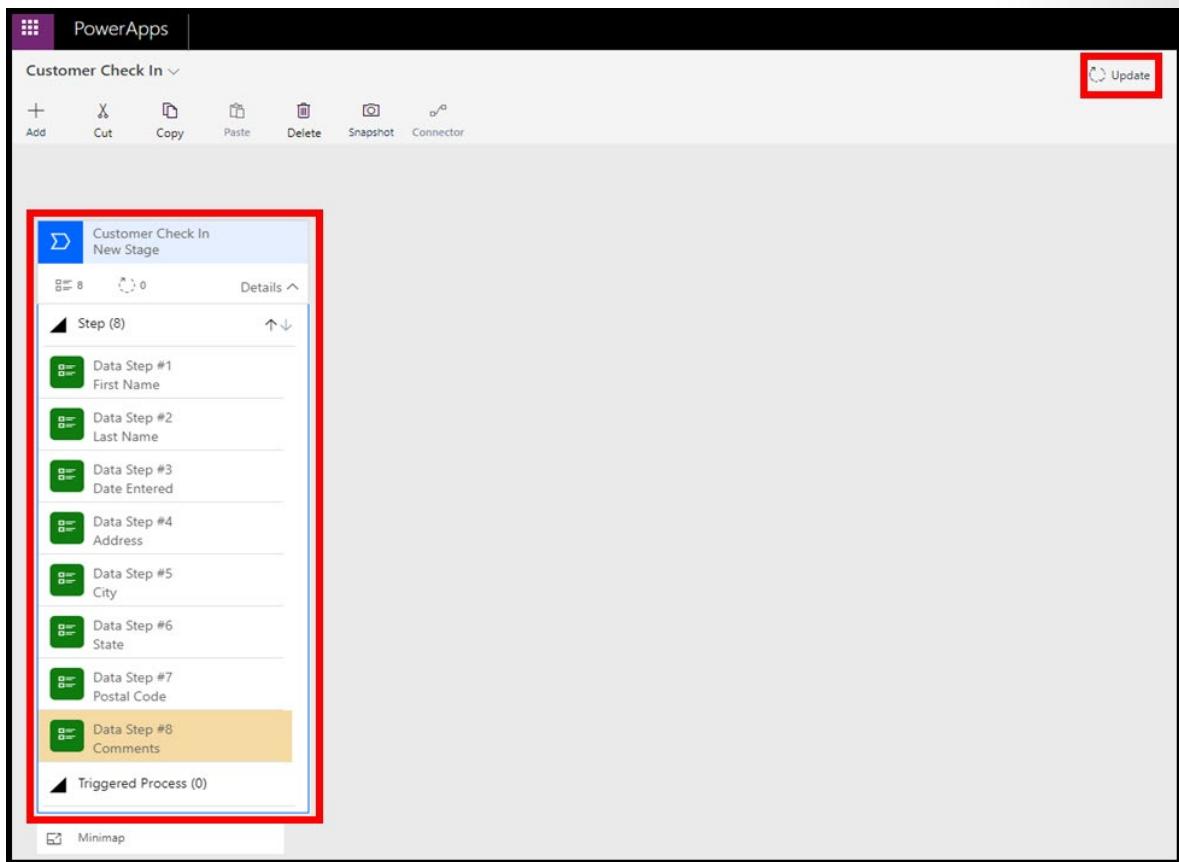


5. Select **Data Step #2** and enter the following information:

- Step Name - Enter **Last Name**.
- Data Field - Select **Last Name** from the drop-down menu.
- Select the **Required** check box.
- Select the **Apply** button.



6. Select the **Components** tab, and then drag additional data steps and add columns that were added earlier under Stage One (Entry Date, Address, City, State, and so on).
7. When you are done, Stage 1 should appear as shown in the following figure. If all appears correct, select the **Update** button in the ribbon.

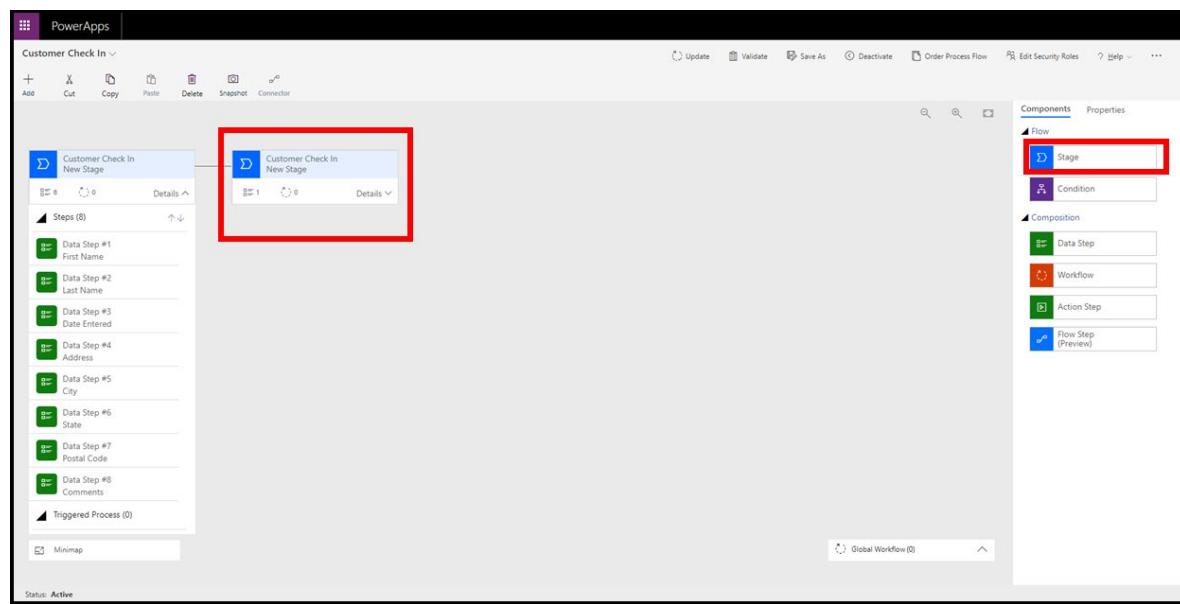


*Note:*

When the data process flow is used, data will be saved into the columns that you created in the customercheckin table in Dataverse.

Now, you will collect information about the automobile being serviced, so you'll add a new stage and add columns about the automobile to the customercheckin table.

8. Select the **Components** tab and drag a new stage to the right of Stage 1. Make sure to drop the new stage into the plus (+) sign. The new stage should resemble the following screenshot.

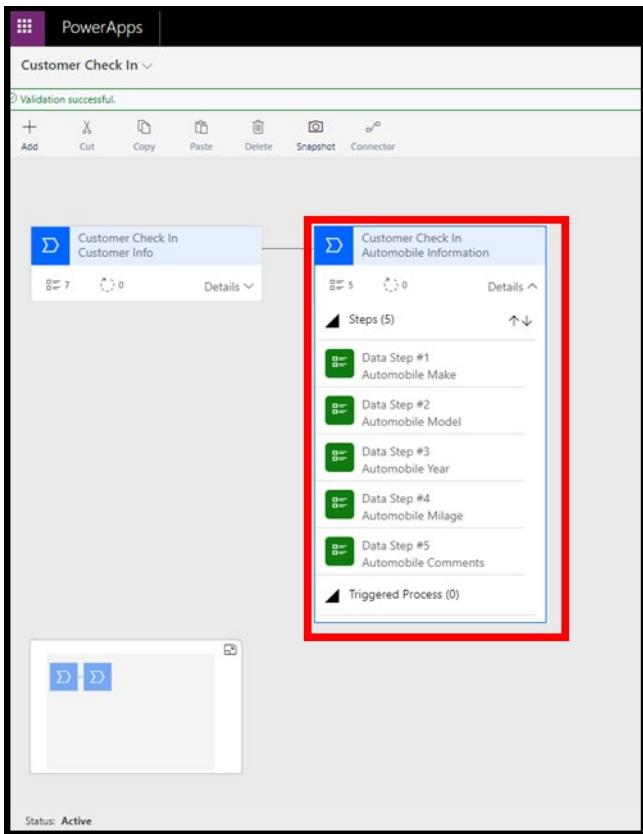


9. Repeat the steps that were covered for Stage 1, and then add the following columns and steps:

- Automobile Make - Text
- Automobile Model - Text
- Automobile Year - Number
- Automobile Mileage - Number
- Automobile Comments - Text Area

10. Make sure to save the table after you add the new columns.

11. When you are done adding the data steps, select the **Update** button in the ribbon. Your completed Stage 2 should look like the following screenshot.



## Exercise - Run the business process flow and view data

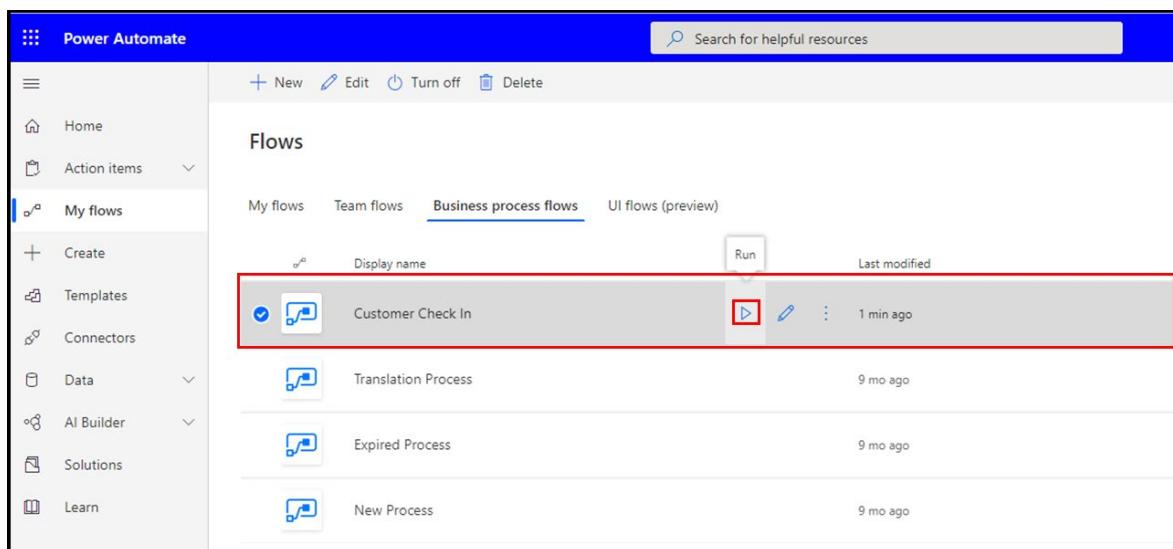
Now that you have a new business process flow, you can try it out and discover how the data is stored after it has been run.

1. Select the **Update** button in the top ribbon.
2. Close the business process flow designer after you see the message that the process flow has updated successfully.
3. Go back to **My Flows**. Select **Business process flows**, where you should see the new flow listed.

*Note:*

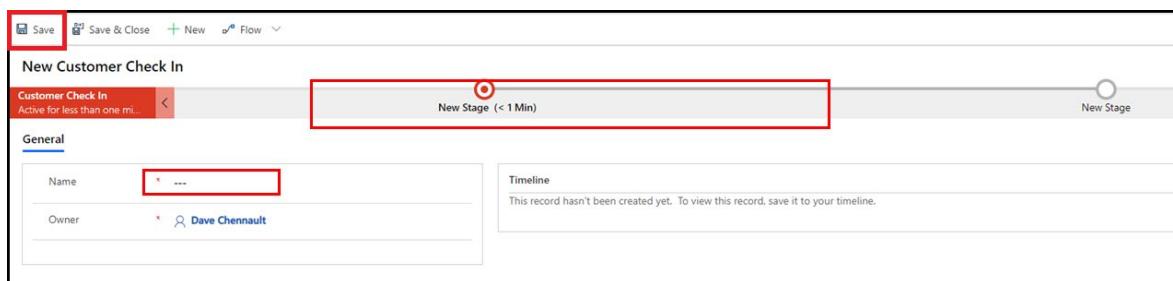
You can see all immersive business process flows that are in process or that have been run by going to the main Power Automate page, selecting the **My flows** icon on the left menu, and then selecting the **Business process flows** tab.

4. Select the **Run** arrow, as shown in the following screenshot.

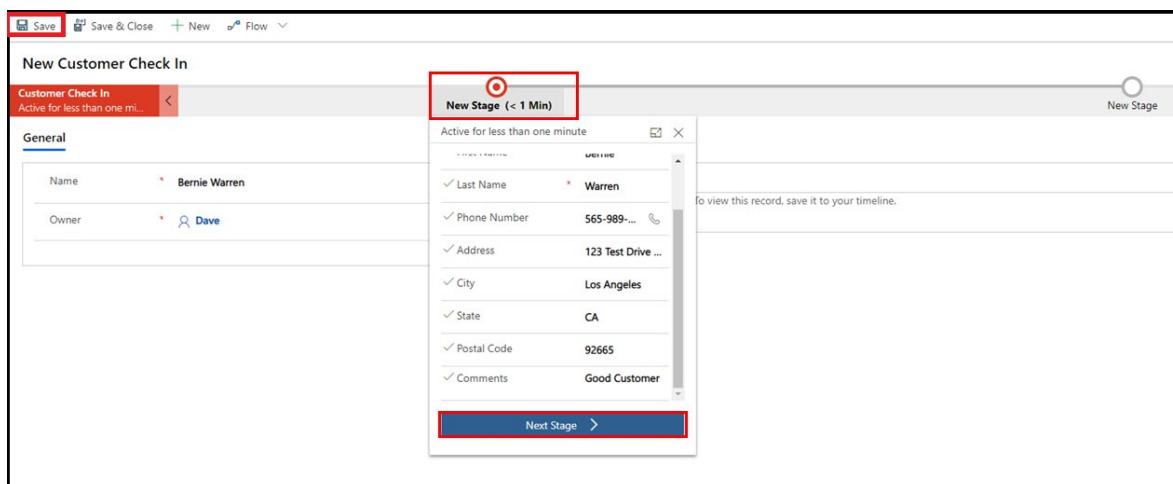


This selection will launch a new instance of the flow that you created.

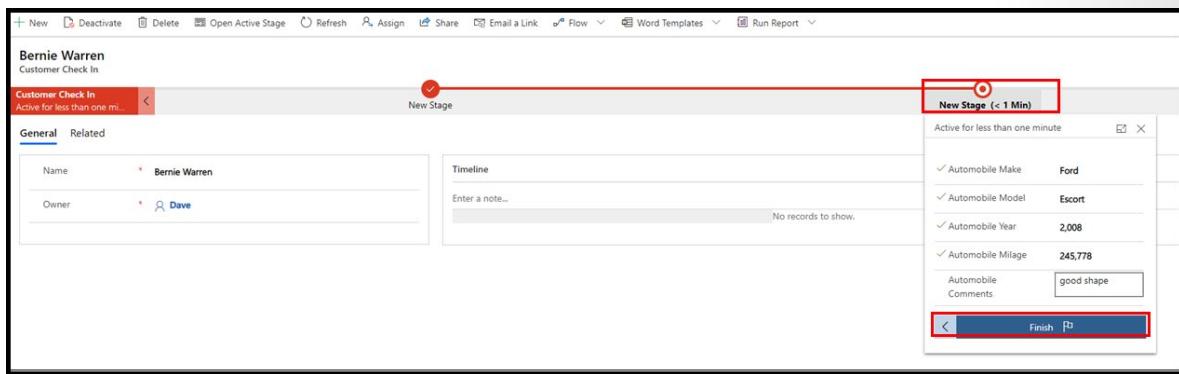
5. Enter a name in the **Name** field on the **General** form (main form) and then select **Save**, as shown in the following figure.



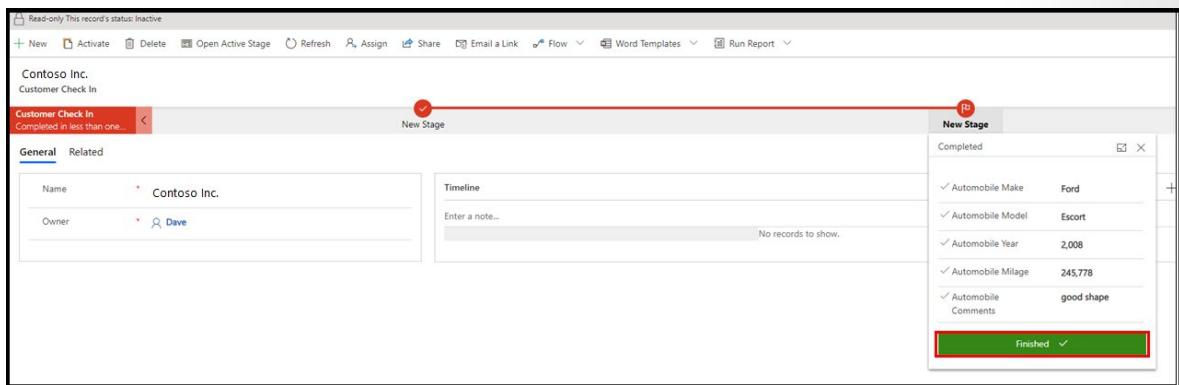
6. After you select the **Save** button, select the red circle for the first stage, fill out the information, and then select the **Next Stage** button.



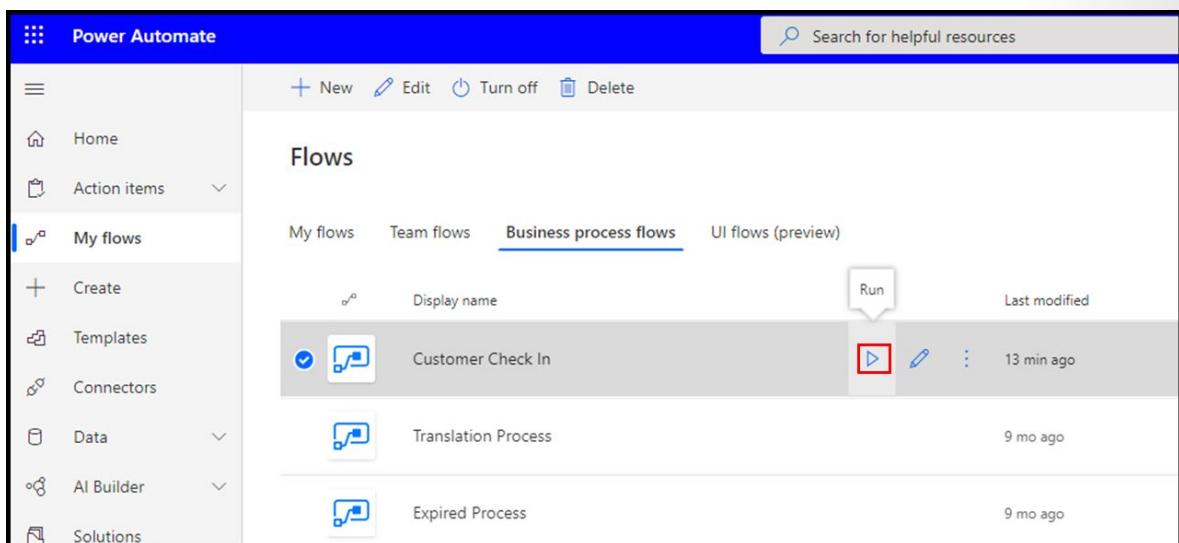
7. Select the second stage, fill out the vehicle information, and then select the **Finish** button.



After you select the **Finish** button, the button in Stage 2 should turn green and the data that you entered is automatically saved.



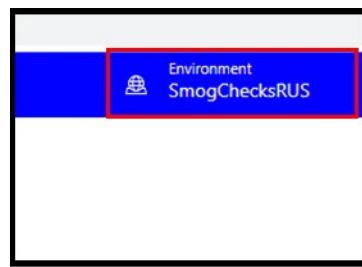
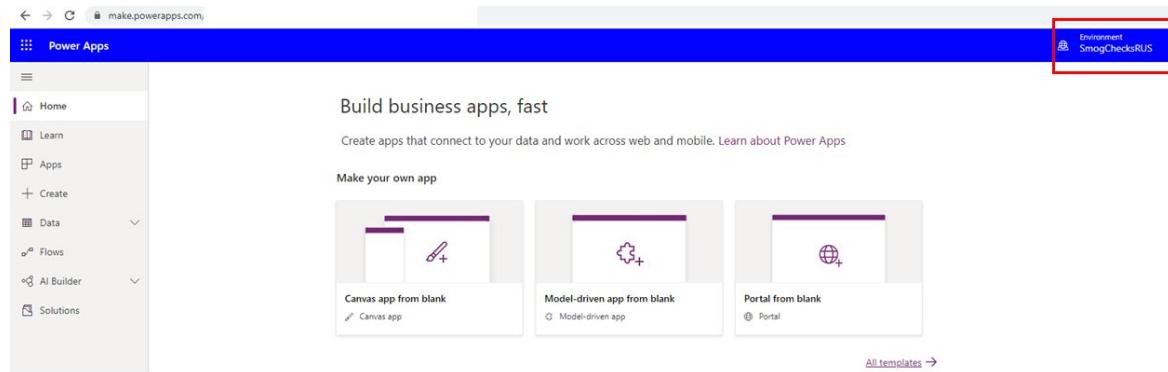
8. You can create additional rows by launching Power Automate, selecting **My Flows** and **Business process flows**, and then selecting the arrow next to the **Customer Check In** business process flow, as shown in the following screenshot.



## View the created data

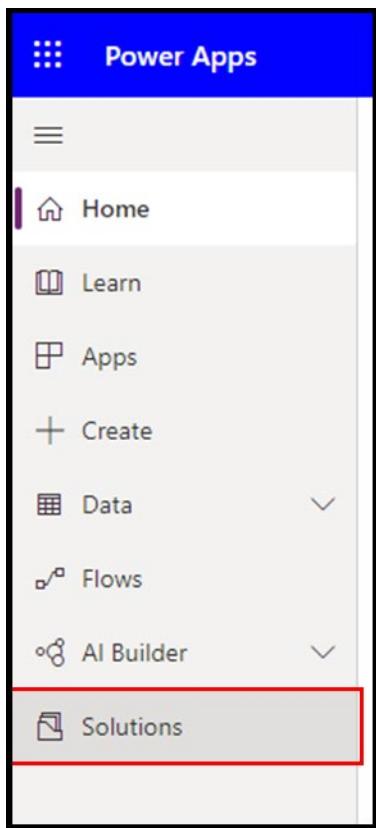
You can view the data that you created for running the flow by following these steps:

1. Go to **Power Apps**<sup>24</sup> and sign in.
2. Select the environment that you used to create the **Customer Check In** business process flow, as shown in the following screenshot.



3. Select the **Solutions** menu on the left side of the screen.

<sup>24</sup> <https://make.powerapps.com/?azure-portal=true>



- Double-click the **Dataverses Default Solution** to open it, as shown in the following figure.

| Solutions                             |                       |           |          |  |
|---------------------------------------|-----------------------|-----------|----------|--|
| Display name                          | Name                  | Created   | Version  |  |
| AI Sample Data                        | msdyn_AIBuilderSam... | 1/11/2020 | 1.0.0.5  |  |
| Asset Checkout                        | AssetCheckout         | 1/11/2020 | 0.0.0.1  |  |
| Innovation Challenge                  | InnovationHub         | 1/11/2020 | 0.0.0.1  |  |
| Fundraiser                            | msdyn_SampleApp       | 1/11/2020 | 1.0.0.2  |  |
| Contextual Help Base                  | msdyn_ContextualH...  | 1/11/2020 | 1.0.0.10 |  |
| Contextual Help                       | msdyn_ContextualH...  | 1/11/2020 | 1.0.0.10 |  |
| Common Data Services Default Solution | Crd83b3               | 1/11/2020 | 1.0.0.0  |  |
| Default Solution                      | Default               | 1/11/2020 | 1.0      |  |

- Select the **Customer Check In** table.

Solutions > Common Data Services Default Solution

| Display name                                          | Name                  | Type       | Managed by |
|-------------------------------------------------------|-----------------------|------------|------------|
| Author                                                | cr5be_author          | Entity     |            |
| Boat Name                                             | cr5be_boatname        | Option Set |            |
| Books                                                 | cr5be_books           | Entity     |            |
| Captain                                               | cr5be_captain         | Option Set |            |
| Cruises                                               | cr5be_cruises         | Entity     |            |
| Destination                                           | cr5be_destination     | Option Set |            |
| Dogs                                                  | cr5be_dogs            | Entity     |            |
| Gender                                                | cr5be_gender          | Option Set |            |
| Identification                                        | cr5be_identification  | Option Set |            |
| Library Book                                          | cr5be_librarybook     | Entity     |            |
| Passengers                                            | cr5be_passengers      | Entity     |            |
| PC                                                    | cr5be_pc              | Entity     |            |
| PC Type                                               | cr5be_pctype          | Option Set |            |
| Screen Size Field                                     | cr5be_screensizefield | Option Set |            |
| Sex                                                   | cr5be_sex             | Option Set |            |
| ID Required                                           | ID Required           | Process    |            |
| <input checked="" type="checkbox"/> Customer Check In | new_customercheckin   | Entity     |            |

6. Select the **Data** tab then click **Select view** on the right side of the screen, and then select the **All Data** view option.

Solutions > Common Data Services Default Solution > Customer Check In

Fields Relationships Business rules Views Forms Dashboards Charts Keys Data

| Name           | Customer Check In    | Active Stage | Active Stage Star... | Address            | Automobile Com... | Automobile Make | Automobile Mil... | Automobile Model | Automobile Year | City          | Comm     |
|----------------|----------------------|--------------|----------------------|--------------------|-------------------|-----------------|-------------------|------------------|-----------------|---------------|----------|
| Joe Green      | 0791c597-3237-ea1... | New Stage    | 1/14/2020            | 123 West End Way   | Beater            | Chevrolet       | 676,009           | Nova             | 1,977           | San Remo      | Great g  |
| William Sonoma | 30cc4656-3337-ea1... | New Stage    | 1/14/2020            | 123 East End Drive | Wow!              | Ferrari         | 3,600             | 333              | 2,010           | San Francisco | Nice ca  |
| William Wilson | 63a169de-bb37-ea1... | New Stage    | 1/15/2020            | 5551 West End Way  | Nice car          | Toyota          | 165,779           | Camry            | 2,008           | Laguna Beach  | repeat i |
| William Tell   | 3225f62a-ca37-ea1... | New Stage    | 1/15/2020            | 334 Rose Lane      |                   |                 |                   |                  |                 | Huntington    |          |
| Bernie Warren  | 8bde3304-cd37-ea1... | New Stage    | 1/15/2020            | 123 Test Drive Way | good shape        | Ford            | 245,778           | Escort           | 2,008           | Los Angeles   | Good C   |

| Address            | Automobile Company | Automobile Make | Automobile Miles | Automobile Model | Automobile Year | City          | Comm     |
|--------------------|--------------------|-----------------|------------------|------------------|-----------------|---------------|----------|
| 123 West End Way   | Beater             | Chevrolet       | 676,009          | Nova             | 1,977           | San Remo      | Great g  |
| 123 Est End Drive  | Wow!               | Ferrari         | 3,600            | 333              | 2,010           | San Francisco | Nice ca  |
| 5551 West End Way  | Nice car           | Toyota          | 165,779          | Camry            | 2,008           | Laguna Beach  | repeat c |
| 334 Rose Lane      |                    |                 |                  |                  |                 | Huntington    |          |
| 123 Test Drive Way | good shape         | Ford            | 245,778          | Escort           | 2,008           | Los Angeles   | Good C   |

Now, you can view the all the data that you created with your new immersive business process flow.

## Summary

This module explained the tools that you can use for building your first immersive business process flow. In addition, you learned about the following concepts:

- The definition of an immersive business process flow.
- The differences between an embedded and immersive business process flow.
- How to create a new immersive business process flow in Power Automate.
- How to add columns to an table by using the business process flow designer.
- How to build and save a two-step business immersive business process flow by using stages and data steps.
- How to run an immersive business process flow.
- How to view the data that you created in the new immersive business process flow.

Though immersive business process flows are the main focus of this module, business process flows can also be created and embedded within a model-driven application. If you want to learn more about embedded business process flows, select the following links for a series of videos that demonstrate the building of an embedded business process flow within a model-driven app. The videos are informative and led by Microsoft Power Automate Product Managers.

[Introduction & Planning a Business Process Flow<sup>25</sup>](#)

[Building a Model Driven App<sup>26</sup>](#)

[Build A Business Process Flow and Add to Model Driven Apps<sup>27</sup>](#)

[Add a Flow to help manage Stages in a Business Process Flow<sup>28</sup>](#)

<sup>25</sup> <https://www.youtube.com/watch?v=7RiXDIPNZic>

<sup>26</sup> <https://www.youtube.com/watch?v=sslyrDVCaw8>

<sup>27</sup> [https://www.youtube.com/watch?v=e4u9fE\\_teNo](https://www.youtube.com/watch?v=e4u9fE_teNo)

<sup>28</sup> <https://www.youtube.com/watch?v=9DFgFax0lBo>

**Watching the Business Process Flow in Action<sup>29</sup>**

---

**29** <https://www.youtube.com/watch?v=X0sjXE33oGM>

# Introduction to Power Automate security and governance

## Introduction

Microsoft Power Platform provides organizations with the ability to automate business processes and create apps by using an intuitive drag-and-drop designer. The audience for Microsoft Power Platform is diverse and is often referred to as makers.

Whether you are an information worker who is familiar with Microsoft Excel or a professional developer who is building an application in Node.js, Microsoft Power Platform can be used to improve productivity through a low code experience.

A powerful capability in Microsoft Power Platform is connectors. By using connectors, a maker can connect to a wide variety of applications and APIs without having to understand security schemes, such as OAuth, to establish a connection to these systems. Currently, more than 300 connectors are provided by Microsoft and third-party organizations.

With democratized access to building flows and apps, organizations might have concerns with makers connecting to unauthorized systems, which could result in the leaking of sensitive business information. For many organizations, balancing increased productivity comes with additional risks and exposure. However, Microsoft provides tools and guidance that organizations can use to find the right balance between digital transformation, allowing them to be more competitive in their industry without exposing sensitive data.

## Identify Power Platform environments

Environments are an important consideration when you are trying to help secure and govern Power Apps and Power Automate usage. Environments act as a security container for apps and flows to run within. Every flow must be assigned to an environment. Every licensed user belongs to the default environment.

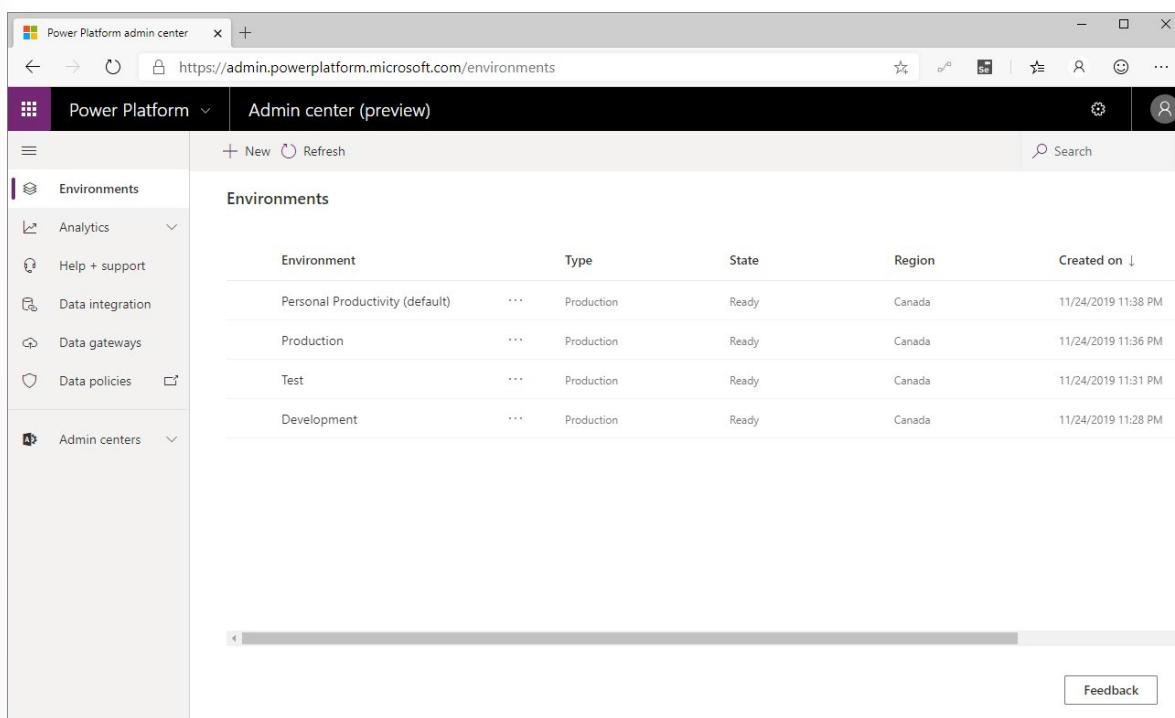
Additional environments can be provisioned where opt-in access is possible. A strategy that some organizations adopt includes **renaming the default environment<sup>30</sup>** to be called **Personal Productivity**, which implies permission to build flows that improve their own productivity.

Organizations can subsequently create additional environments for development, testing, and production purposes. Using this strategy might also align flow creation with existing IT change management requirements.

By default, anyone can create environments, but administrators can also **control who can create and manage environments in the Power Platform Admin center<sup>31</sup>** and limit it to a set of administrators.

<sup>30</sup> <https://docs.microsoft.com/power-platform/admin/environments-administration?azure-portal=true#rename-your-environment>

<sup>31</sup> <https://docs.microsoft.com/power-platform/admin/control-environment-creation/?azure-portal=true>



The screenshot shows the Microsoft Power Platform admin center interface. The left sidebar has a 'Environments' section expanded, showing 'Analytics', 'Help + support', 'Data integration', 'Data gateways', 'Data policies', and 'Admin centers'. The main content area is titled 'Environments' and lists four environments: 'Personal Productivity (default)', 'Production', 'Test', and 'Development'. Each environment row includes columns for 'Environment', 'Type', 'State', 'Region', and 'Created on'. The 'Created on' column is sorted by date, with the most recent entry at the top.

| Environment                     | Type | State      | Region | Created on                    |
|---------------------------------|------|------------|--------|-------------------------------|
| Personal Productivity (default) | ...  | Production | Ready  | Canada<br>11/24/2019 11:38 PM |
| Production                      | ...  | Production | Ready  | Canada<br>11/24/2019 11:36 PM |
| Test                            | ...  | Production | Ready  | Canada<br>11/24/2019 11:31 PM |
| Development                     | ...  | Production | Ready  | Canada<br>11/24/2019 11:28 PM |

When creating an environment, organizations can choose which region they want their environment to reside in. This approach is important because it allows organizations to store data closer to actual users and to maintain and meet compliance requirements for their geography. Regions already available to store data include Asia, Australia, Canada, Europe, France, India, Japan, South America, United Kingdom, United States, and US Government (GCC).

Regions are also important when it comes to Admin Analytics because the Power Platform Analytics feature isolates analytics through environments. The telemetry that is generated in one region is not allowed to leave that region. For administrators to view analytics, they need to select an environment first before they can view telemetry.

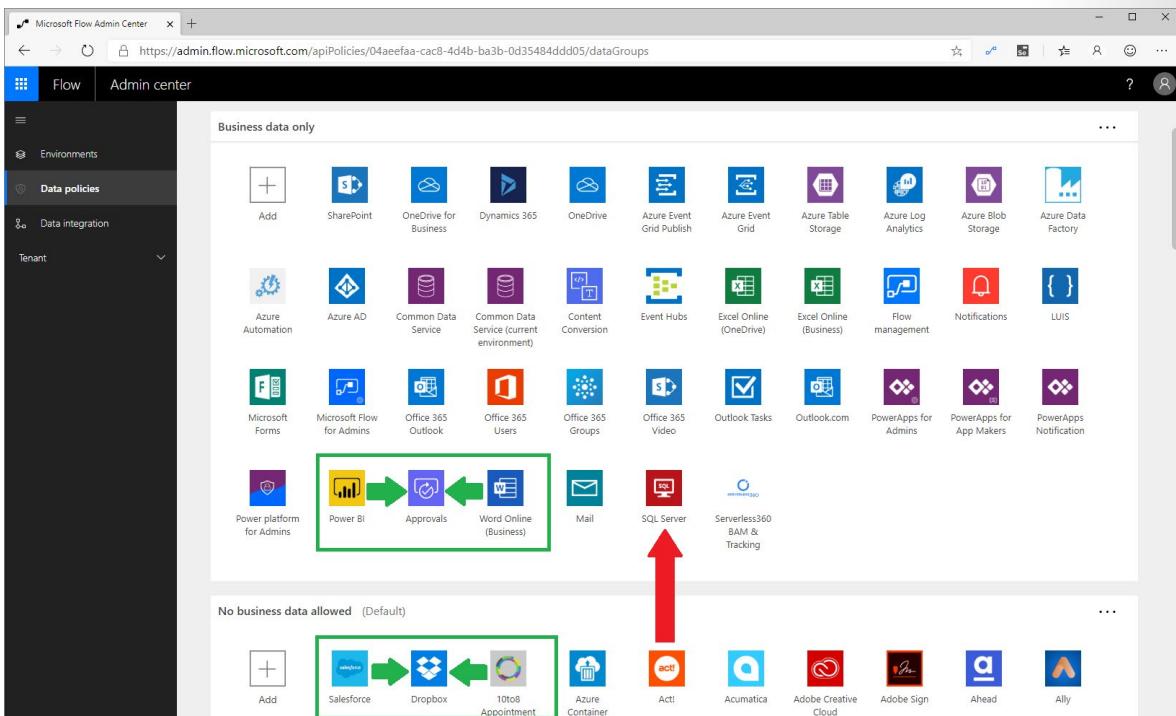
## Data Loss Prevention policies

Data Loss Prevention (DLP) policies allow organizations to construct rules that permit and prevent connectors from communicating with each other within the same flow. Microsoft accomplishes this task by establishing two data groups: **Business data only** and **No business data allowed**.

The intent of these two data groups is to put connectors that have the same data profile in the same data group. Grouping connectors means that they can communicate with each other within the same data group but are not able to communicate across data groups.

To illustrate this concept, consider the following scenario where connectors have been distributed across the two data groups. You want to build a flow that includes the Microsoft **Word Online**, **Approvals**, and **Power BI** connectors. Because all of these connectors belong to the **Business** data group, you can confidently create and run this flow.

Similarly, in the **Non-business** data group, you can combine the **Salesforce**, **Dropbox**, and **10to8 Appointment** connectors within the same flow because they all belong to the same data group. However, you cannot create a flow that includes the **SQL Server** connector and **Act!** connector because they belong to different data groups. If you want to create a flow that included these two connectors, you will need to move one of the connectors into the other data group.

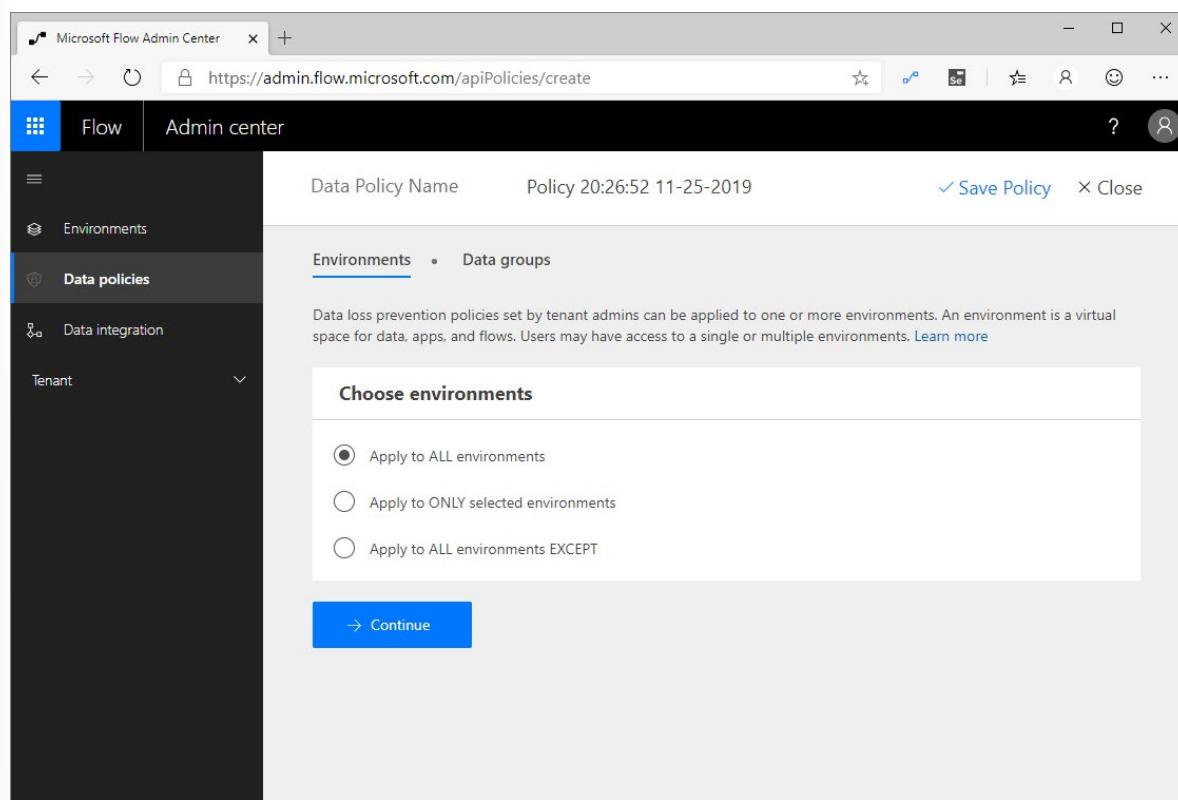


Additionally, you can specify one data group to be the default data group. Select the data policy you want to make default and choose the **Edit policy** button at the top. To change the default data group, go to **Connectors** and choose the **Set default group** button in the upper-right corner.

The default data group is significant because it becomes the data group that all new connectors are added to. For example, if Microsoft deploys a new connector to the Power Automate service, that connector will be added to the default data group. This behavior is also true for custom connectors that are deployed to an environment.

DLP policies can belong to two different scopes: environment and tenant. An environment-scoped DLP policy only applies to that specific environment, whereas a tenant-scoped DLP policy applies to all environments in that tenant. When creating a new DLP policy, policy authors have a few options that they can choose from, including the ability to **Add all environments** (tenant), **Add multiple environments**, and **Exclude certain environments**.

When a DLP author selects **Add all environments**, this setting also ensures that new environments have this DLP policy applied.



Environments can have multiple DLP policies applied, which provides for more governance scenarios to be implemented. However, when multiple DLP policies have been implemented, the most restrictive policy is applied. DLP policy layering is discussed further in the **Planning** module.

Configuring DLP policies depends on an organization's design and cyber security principles. For some organizations, these principles allow for the mixing of business-related connectors with consumer-based services. Other organizations might choose to strictly prevent business-related connectors from connecting with consumer-based services. Organizations should begin by cataloging the business systems that exist within their organization and then creating DLP policies that align with that business mapping.

Organizations should also carefully evaluate which data group they would like to declare as their default data group. Regardless of which data group they designate, administrators should pay attention to new connectors that are being deployed into environments so they can place them into the appropriate data group.

## Power Platform Center of Excellence Starter Kit

Microsoft Power Platform Center of Excellence (COE) Starter Kit is a set of apps, flows, a custom connector, and a Power BI dashboard that allows organizations to govern their Microsoft Power Platform environments. The tool is

---

freely available for download on  
[GitHub<sup>32</sup>](#).

The Starter Kit is a great tool for organizations to increase the visibility of what their makers are doing in their tenant. Naturally, this tool immediately applies to two personas: Cyber Security analysts and Organizational Change Management (OCM) interests. Makers that are building applications and services that introduce risk to the organization should be monitored, but makers who are automating workloads within sanctioned systems and services should be empowered and encouraged to expand on their efforts.

The Starter Kit does have some prerequisites, including:

- A global tenant admin, Microsoft Power Platform Service admin, or Dynamics 365 service admin role is required to access the tenant resources.
- A Power Apps premium license for accessing Microsoft Dataverse.
- An environment with Dataverse because the Starter Kit solution will need to store metadata about the apps and flows that are detected within an environment.
- Power BI Desktop to view the reports and visualizations that highlight app and flow use within the tenant.

Within the Starter Kit, administrators will discover the tools and features that are discussed in the following sections.

## DLP Strategy

By using the DLP Editor app, an administrator can explore existing DLP policies and evaluate the impact of moving a connector from one data group to another. If a change to a data group has an impact on an existing app, that will be highlighted in the **Affected Power Apps** list. Then, an administrator can send an email, through an in-app experience, to the owner of that application and warn them of the upcoming change.

This current, in-product DLP editor experience doesn't provide this type of *what-if* analysis, so we recommend that you use the DLP Strategy app that comes with the COE Starter Kit when making DLP changes.

Using the DLP Strategy app will also help you understand the impact to existing apps when DLP changes are made and help you reduce the chance of unknowingly breaking someone's app.

---

<sup>32</sup> <https://github.com/microsoft/powerapps-tools/tree/master/Administration/CoEStarterKit/?azure-portal=true>

The screenshot shows the Microsoft Power Platform Admin center interface. On the left, there's a navigation bar with 'DLP Editor' selected. The main area is titled 'Playground Policy'. It has two main sections: 'BUSINESS DATA ONLY (9)' and 'NON-BUSINESS DATA (252)'. Under 'Affected PowerApps (1 of 12)', it lists 'App Creator Portal' from 'Contoso (Development) (cont...)'. To the right, there's a sidebar for 'APP CREATOR PORTAL' showing details for Megan Bowen, last modified on April 24, 2019, with the app ID cbc055d-3f6e-478f-a7e7-3aabf8aa0a... The sidebar also shows 'Business data only' and 'Non-business data' sections.

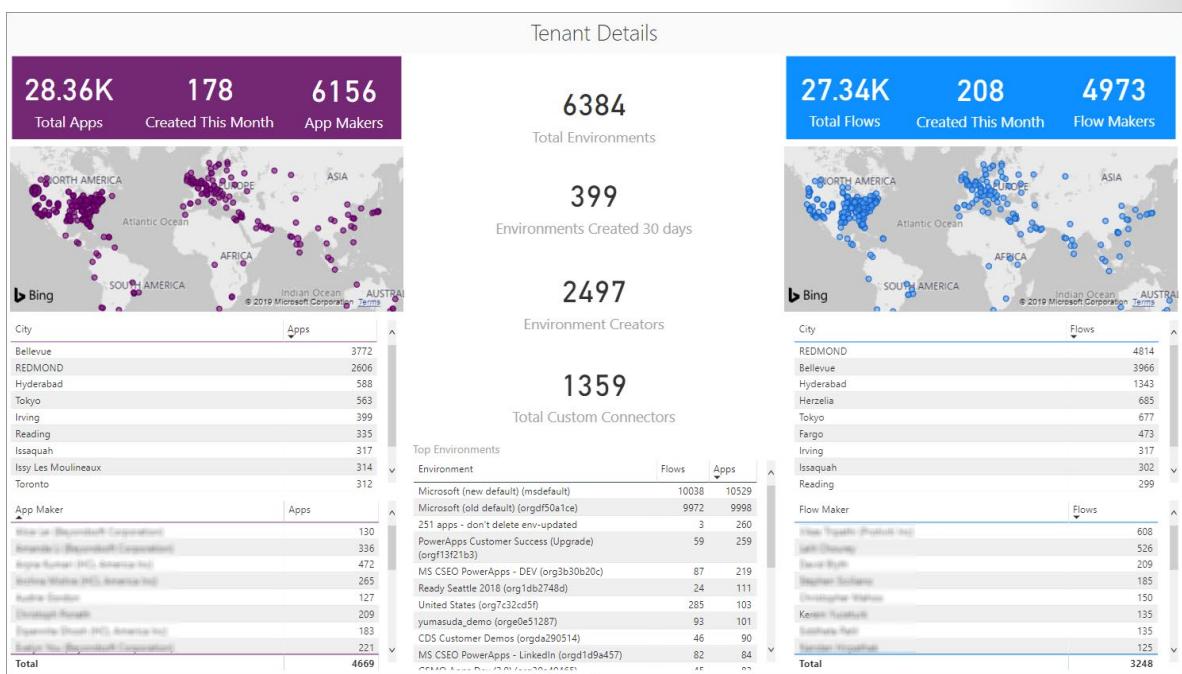
## Catalog tenant resources, visualize data in Power BI

While Admin Analytics, which is found in Microsoft Power Platform Admin center, provides some insight into the use of Power Apps and Power Automate, it does so within the context of an environment. This parameter makes developing a holistic picture of usage difficult. However, using the tools that are provided in the COE Starter Kit brings all these insights together by using Dataverse, a model-driven Power App, and a Power BI dashboard. These assets, plus the inclusion of flows that will sync data from Office 365 Security and Compliance logs and through Microsoft Power Platform management connectors into Dataverse entities, provide the greatest level of visibility for administrators.

The Power BI dashboard connects to Starter Kit Dataverse entities and contains tabs that provide the following reports:

- **Overview** - Includes the total number of apps and flows that have been created, including a breakdown based on the maker's location and the number of apps/flows that they have created.
- **Environments** - Highlights the number of environments that have been created and includes the type of environment (Default, Production, Sandbox, Trial), and then identifies the creators of the environments and when the environment was created.
- **Apps** - Provides insights into the usage of apps, including the number of sessions and users that the app has been shared with. This report also breaks down usage by department by pivoting data based on the department that has been specified inside of a user's Office 365 profile.

- **App detail** - Examines the specifics of the app, including the connection type and when the app was last published.
- **Flows** - Displays a flow creation trend that will break down when flows were created. This report will also pivot data based on the user's department and includes a breakdown of the flows that were created by environment.
- **Custom Connectors** - Lists all the custom connectors that exist within a tenant. This report also lists the users that created the connectors and then trends the month that they were created in.
- **Makers** - Identifies all the makers within the environment and identifies how many apps and flows they have built. A slicer exists that allows a consumer of this report to filter based on the maker's department.
- **Connections** - Displays the number of connections that have been established, by connector. This report provides great insight into the popularity of specific connectors within your tenant.

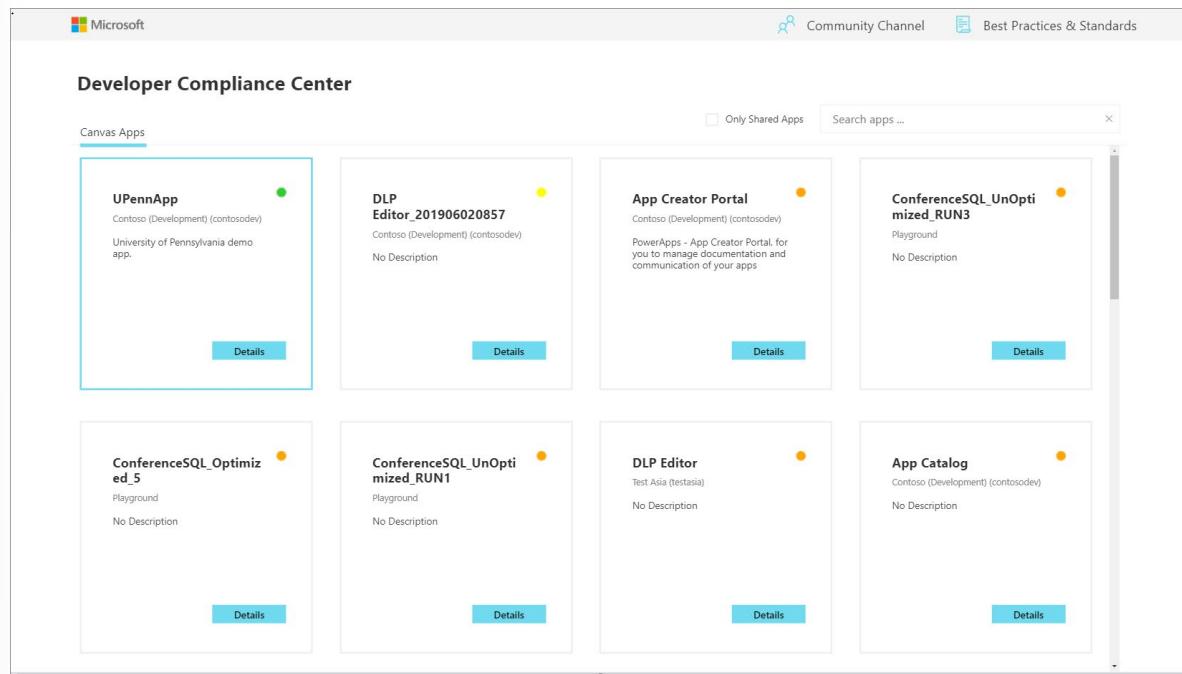


## App Audit

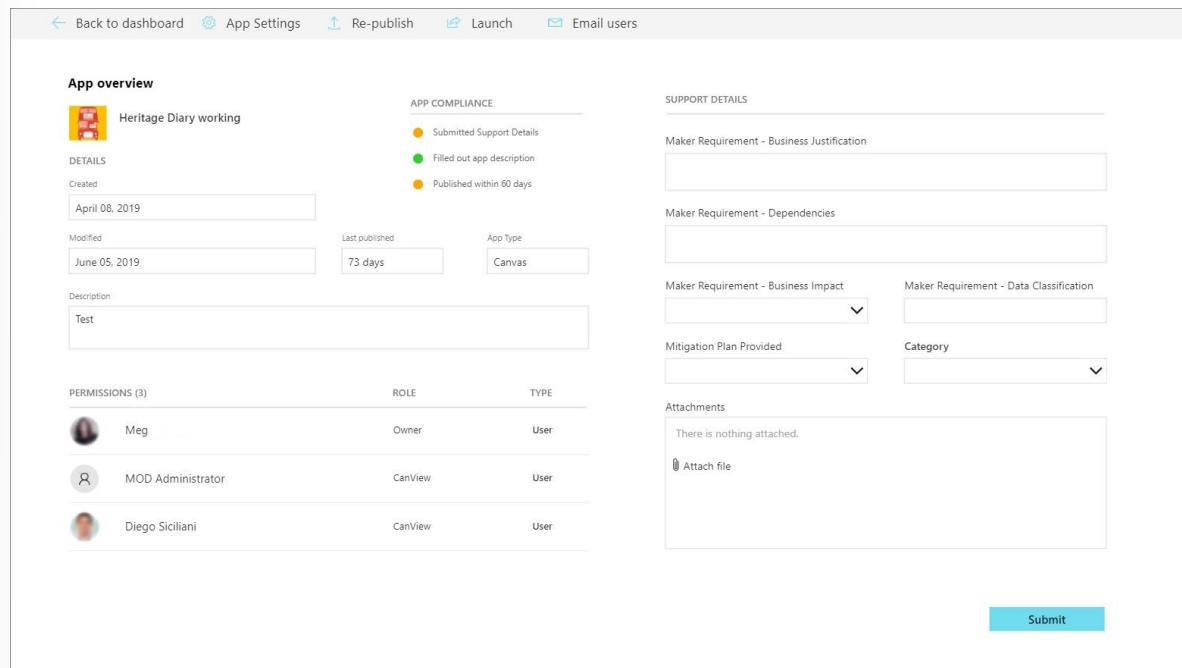
The main purpose of the Sample App Audit process is to demonstrate how an admin could identify overshared or often used resources and gather further information, like business justification and business impact of an outage, for those apps.

Having a platform that provides democratized access to technology is only useful if people don't create many redundant or low-value applications. Situations might also occur where people create temporary applications for proof-of-concept purposes. These apps can clutter an environment if they are not cleaned up. Within the COE Starter Kit, app makers can be prompted to attest their application

to ensure that it addresses business justification requirements. If the application doesn't address these requirements, makers can have their application removed from the environment.



The screenshot shows the Microsoft Developer Compliance Center interface. At the top, there are navigation links for 'Community Channel' and 'Best Practices & Standards'. Below that, a search bar with the placeholder 'Search apps ...' and a checkbox for 'Only Shared Apps'. The main area displays a grid of app cards under the heading 'Developer Compliance Center' and 'Canvas Apps'. Each card includes the app name, developer information, a status indicator (green dot for UPennApp, yellow dot for others), a brief description, and a 'Details' button. The cards are arranged in two rows of four.

The screenshot shows the Microsoft App Catalog interface. At the top, there are navigation links: 'Back to dashboard', 'App Settings', 'Re-publish', 'Launch', and 'Email users'. Below that, the 'App overview' section for the app 'Heritage Diary working'. It includes details like creation date (April 08, 2019), last published date (June 05, 2019, 73 days ago), and app type (Canvas). The 'APP COMPLIANCE' section shows status indicators: green dot for 'Filled out app description', yellow dots for 'Submitted Support Details' and 'Published within 60 days'. The 'SUPPORT DETAILS' section contains fields for 'Maker Requirement - Business Justification', 'Dependencies', 'Business Impact', 'Data Classification', 'Mitigation Plan Provided', and 'Category'. The 'PERMISSIONS (3)' section lists users Meg, MOD Administrator, and Diego Siciliani with their roles (Owner, CanView) and types (User). The 'ROLE' column shows 'Owner' for Meg and 'CanView' for the others. The 'TYPE' column shows 'User' for all. At the bottom right is a 'Submit' button.

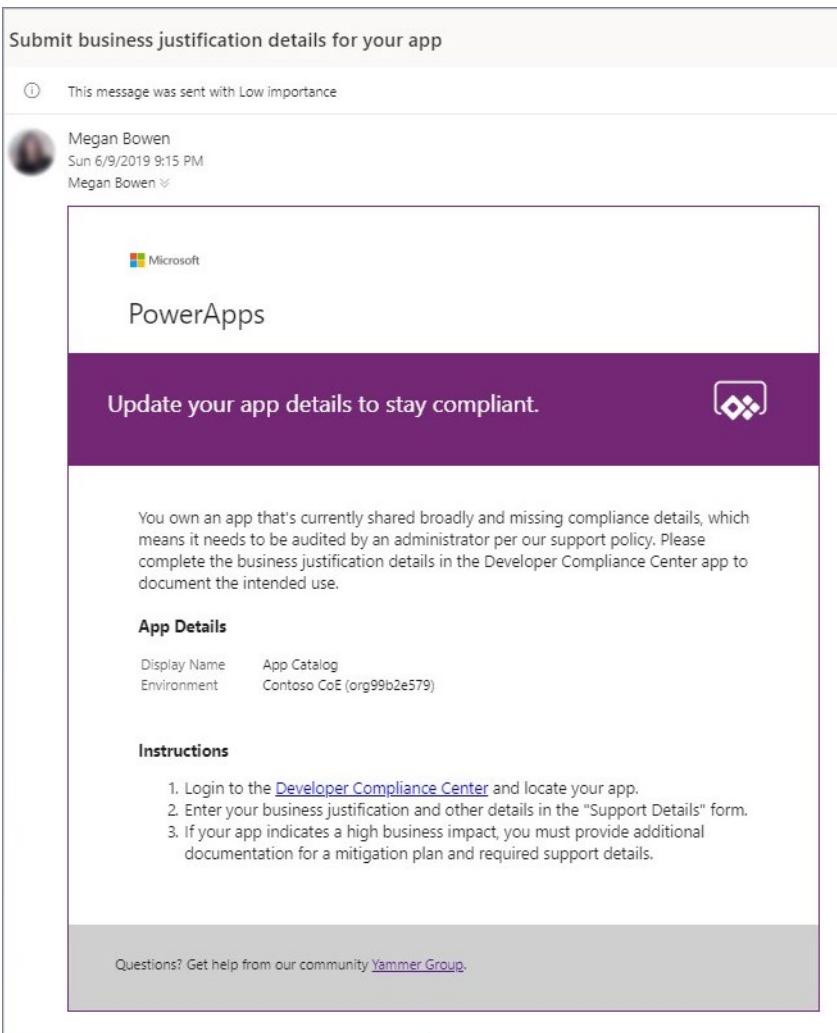
## App Catalog

The App Catalog application acts as a catalog that helps with discoverability of apps. Users can explore featured apps and browse apps by category. The app catalog can be a great entry point to launch apps for end users and makers can explore to see if an application already exists before they create another app that provides similar functionality.

An Admin decides which apps are featured in the App Catalog by completing the **App Audit** process. After an app has been approved by the **App Audit** process, these apps can be featured in the app catalog.

## Set Owner

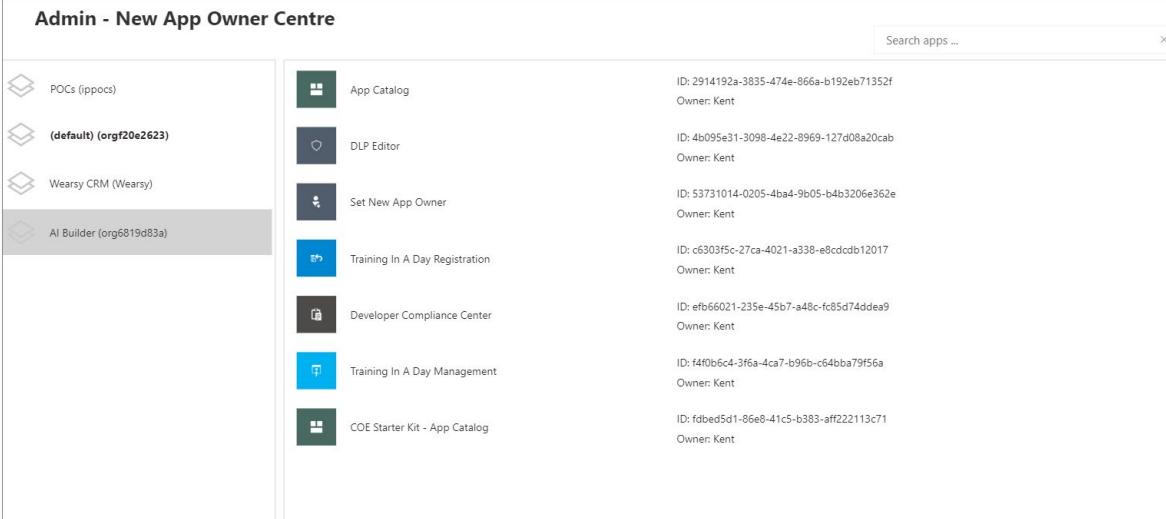
Administrating access to applications can be done by the **Owner** of the application. However, circumstances might occur where you want an administrator to provide access. By using the **Set Owner** app, you can allow for this situation by selecting an app, adding users, and then indicating whether they should be able to view the app or edit the app.



## Welcome email to new makers

Power Apps and Power Automate can be discovered virally within an organization's tenant through the Office 365 portal. Makers might be confused about whether they are allowed to use these tools or not.

Detecting when a maker has created their first app or flow and sending them some useful information to accelerate their journey will immediately imply that these tools are the ones that they are permitted to use. Now, they will have a list of resources that can ease their transition into using these tools.



The screenshot shows the 'Admin - New App Owner Centre' interface. On the left, there's a navigation pane with several items: 'POCs (ippocs)', '(default) (orgf20e2623)', 'Wearsy CRM (Wearsy)', and 'AI Builder (org6819d83a)', with 'AI Builder' currently selected. The main area displays a list of apps with their details:

| App                            | ID                                       | Owner |
|--------------------------------|------------------------------------------|-------|
| App Catalog                    | ID: 2914192a-3835-474e-866a-b192eb71352f | Kent  |
| DLP Editor                     | ID: 4b095e31-3098-4e22-8969-127d08a20cab | Kent  |
| Set New App Owner              | ID: 53731014-0205-4ba4-9b05-b4b3206e562e | Kent  |
| Training In A Day Registration | ID: c6303f5c-27ca-4021-a338-e8cdcd12017  | Kent  |
| Developer Compliance Center    | ID: efb66021-235e-45b7-a40c-fc85d74dde9  | Kent  |
| Training In A Day Management   | ID: f4f0b6c4-3f6a-4ca7-b96b-c64bba79f56a | Kent  |
| COE Starter Kit - App Catalog  | ID: fdbed5d1-86e8-41c5-b383-aff222113c71 | Kent  |

## Exercise - Create a Power Automate environment

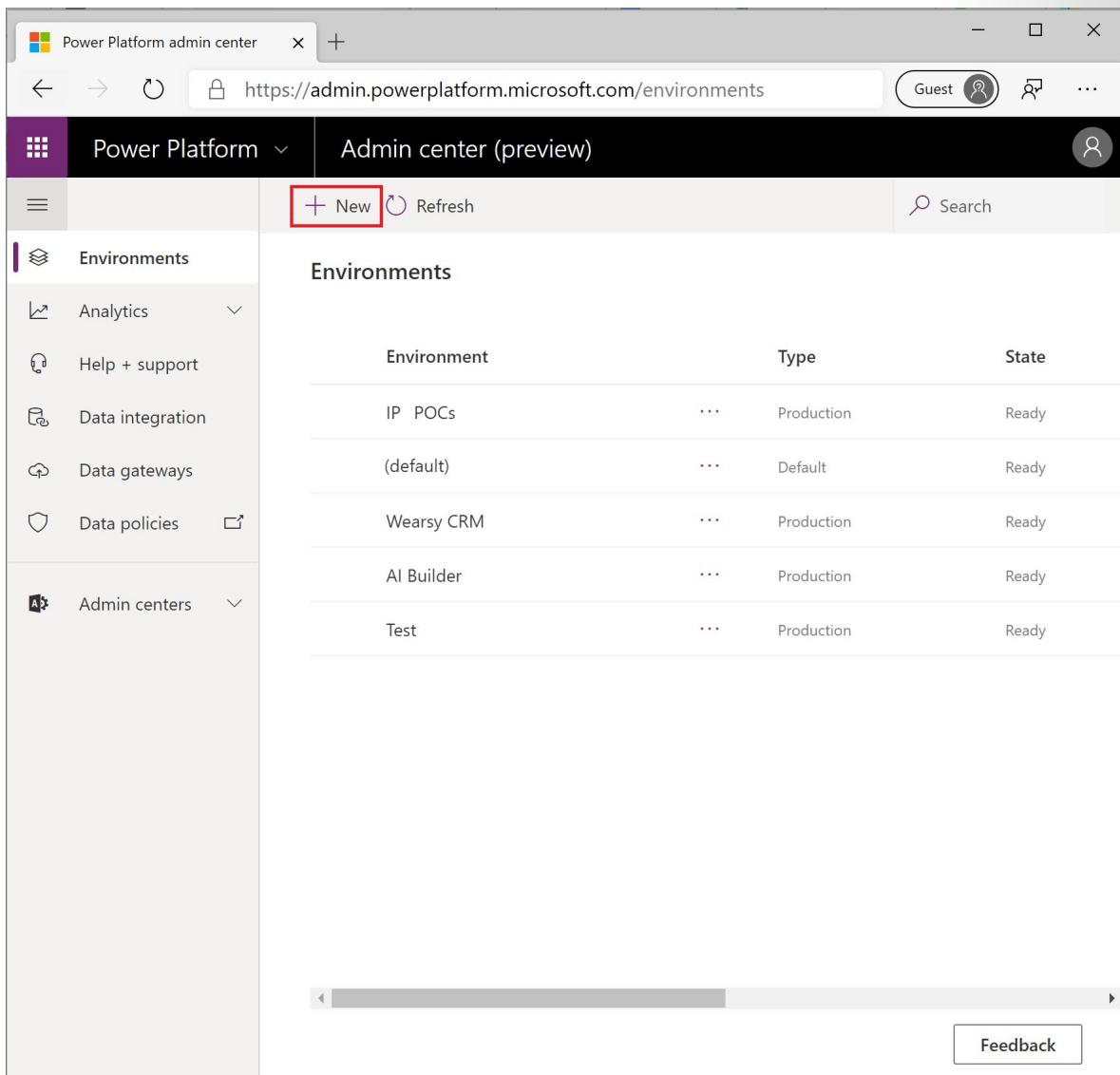
In this exercise, you'll create a Power Automate environment by completing the following steps:

1. To complete this exercise, ensure that your account meets the licensing criteria discussed here in the **Who can create environments<sup>33</sup>** section.  
If necessary, sign up for a **trial Power Automate account<sup>34</sup>**.
2. Sign in to the **Power Platform Admin Center<sup>35</sup>**.
3. Select **Environments** from the left navigation pane.
4. Select **+ New**.

<sup>33</sup> <https://docs.microsoft.com/power-platform/admin/create-environment#who-can-create-environments/?azure-portal=true>

<sup>34</sup> <https://flow.microsoft.com/#home-signup/?azure-portal=true>

<sup>35</sup> <https://admin.powerplatform.microsoft.com/?azure-portal=true>



The screenshot shows the Microsoft Power Platform Admin Center interface. The left sidebar has sections for Environments (Analytics, Help + support, Data integration, Data gateways, Data policies), Admin centers, and a search bar. The main area is titled 'Environments' and lists existing environments: IP POCs (Production, Ready), (default) (Production, Ready), Wearsy CRM (Production, Ready), AI Builder (Production, Ready), and Test (Production, Ready). At the top right, there's a '+ New' button, which is highlighted with a red box.

| Environment | Type       | State |
|-------------|------------|-------|
| IP POCs     | Production | Ready |
| (default)   | Production | Ready |
| Wearsy CRM  | Production | Ready |
| AI Builder  | Production | Ready |
| Test        | Production | Ready |

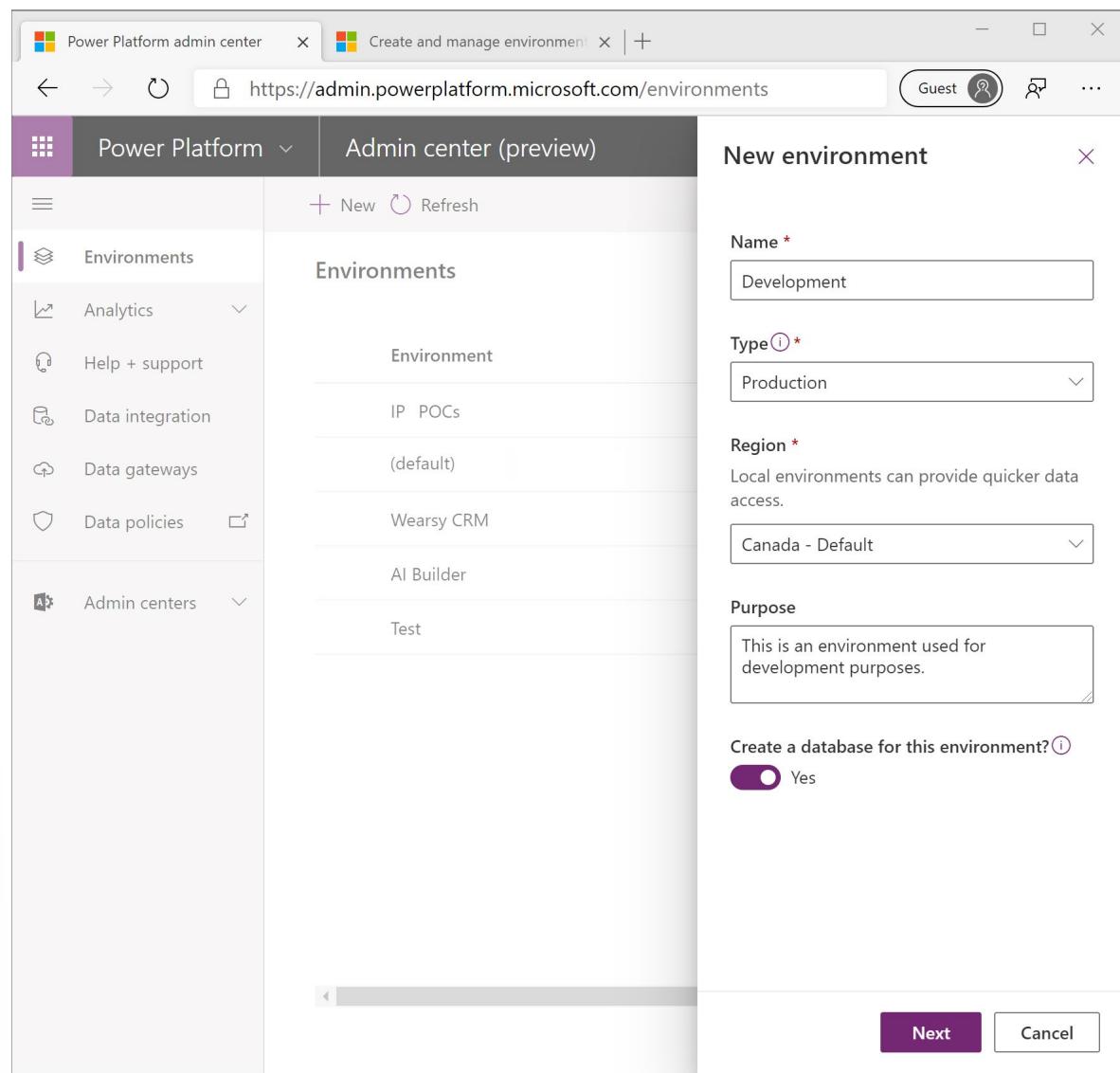
5. Provide a **Name of Development** and then select a **Type** of environment. Depending on your circumstances, create a **Trial** or **Production** environment. If you plan to dispose of this environment after completing the modules in this learning path, select **Trial**. If you anticipate that this environment will be used beyond 30 days, select **Production**.

*Note:*

When selecting the **Type** of environment, you do have three options: **Sandbox**, **Production**, and **Trial**. Sandbox environments are for building proof-of-concept applications and flows that might need to be reset without recreating the entire environment. Production environments are suited for stable workloads where you have predictable usage. Trial environments will expire after 30 days. These environments are best suited for evaluating technologies.

6. Select an appropriate **Region** for your organization, provide a relevant **Purpose** for this environment, and indicate that you want a **Database** to be created. After you have populated these values, select the **Next** button to proceed.

When deciding to include a database for future environments, evaluate how the environment will be used. For example, if you have plans to use Dataverse, AI Builder, UI flows, or apps that use Dataverse, then you need to include a database in your environment. If you have no plans to use these features, then a database is not required.



7. Provide the **Language** of your choice and **Currency**. For the purposes of this exercise, do not enable Dynamics 365 apps or sample apps and data. Select **Save** to create your environment.
8. After a couple minutes, your environment should be provisioned and you should see the following status message displayed in the Admin center. You might need to select **Refresh** for the **State** field to update.

The screenshot shows the Power Platform Admin Center interface. On the left, there's a navigation sidebar with 'Environments' selected. The main area displays a table of environments:

| Environment        | Type       | State |
|--------------------|------------|-------|
| IP POCs            | Production | Ready |
| (default)          | Default    | Ready |
| Wearsy CRM         | Production | Ready |
| <b>Development</b> | Production | Ready |
| AI Builder         | Production | Ready |
| Test               | Production | Ready |

A green success message at the top right says 'New environment Development has been successfully created.' A red box highlights the 'Development' row in the table.

## Summary

This module introduced many different Power Platform concepts, including environments' Data Loss Prevention (DLP) policies, and reviewed many of the tools that are found in the Center of Excellence (COE) Starter Kit. All these features and tools are key to organizations governing their environments.

Because every organization has different governance needs, these tools can be used selectively to implement the level of governance that is required for your organization. In addition, while these tools do enable governance, they can also be used to identify makers who are building solutions that improve an organization's productivity.

When these makers have been identified, organizations should look to further empower them to extend their app development benefits to more parts of the organization.

# Build your first workflow with Power Automate for Teams

## Introduction

Having business processes that are efficient, cost effective, and reliable is essential in today's business. Automated workflows allow you to do more with fewer manual tasks, saving you time and money. Microsoft Power Automate in Dataverse for Teams automates workflows such as reacting to a new Teams message, emailing alerts when data changes in Dataverse, or performing scheduled tasks. With so much work happening in Microsoft Teams, automating business processes allow your users to be more productive.

To create flows in Dataverse for Teams, you need:

- A Microsoft team with a Dataverse for Teams environment. A team is provisioned in the Dataverse for Teams environment when you create an app or bot in Microsoft Teams for the first time or install an app from the Microsoft Power Apps catalog. The Dataverse for Teams environment is used to store, manage, and share team-specific data, apps, and flows. Each team can have one environment, and all data, apps, bots, and flows that are created with the app from Power Apps inside a team are available from that team's Dataverse for Teams database.
- The Power Apps for Teams app to create and edit flows for a Dataverse for Teams environment. Flows that are created in Power Automate for Teams or [https://flow.microsoft.com<sup>36</sup>](https://flow.microsoft.com) are stored in the default environment and not in Dataverse for Teams.

## Types of workflows that Power Automate can build in Dataverse

The types of workflows that can be created are categorized as automated and scheduled.

### Automated flows

Occasionally, you will have tasks that need to be performed when certain events happen. With Power Automate monitoring for those events, you can have the workflow automatically performed.

Examples of automated flows include:

- When an item in a Teams list is modified, send an email
- Create a task in Planner based on a Microsoft 365 Outlook calendar event
- Start an approval for new documents in Teams

### Scheduled flows

Scheduled flows are launched by time-based events. At a certain interval or a specified time, Power Automate will perform the workflow.

Examples of scheduled flows include:

- Email a list of upcoming calendar events
- Delete OneDrive files older than five days and send an email daily

---

<sup>36</sup> <https://flow.microsoft.com/?azure-portal=true>

- Send an Outlook email from a shared mailbox on a recurring basis

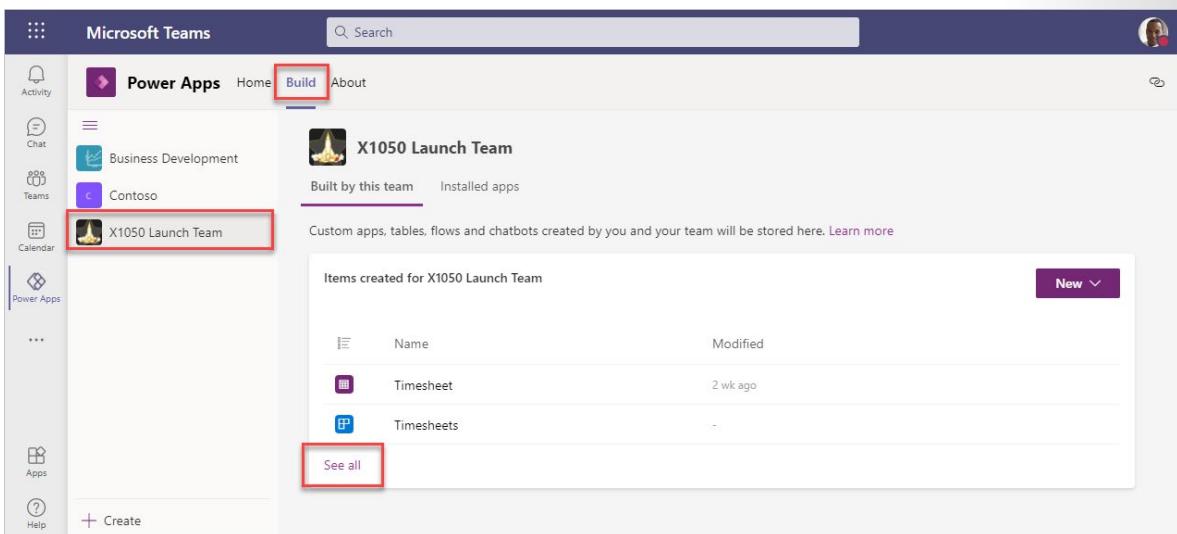
## Exercise-Trigger when a new message is created

In this exercise, you will create your first flow by using Dataverse for Teams. For this example, you'll create an automated workflow that will create a Planner task each time that a message with the word "task" in the subject line is posted to a Teams channel. The flow will need a Planner to create the tasks in. You can identify an existing Planner plan or create a new one.

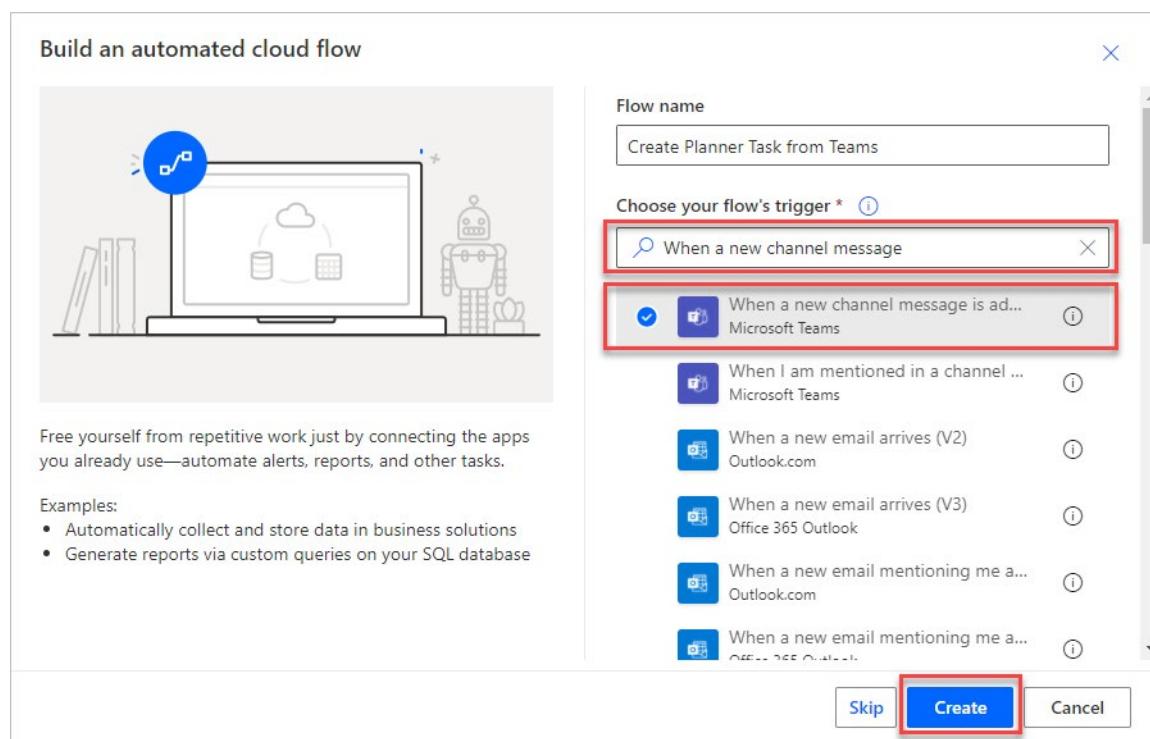
### Step 1: Create a new flow

To create a new flow, follow these steps:

1. From Teams, open the Power Apps application.
2. From the **Build** tab, select the team on the left for the flow.
3. Only teams that have a Dataverse for Teams environment will appear.
3. Under the **Built by this team** tab, select **See all**.



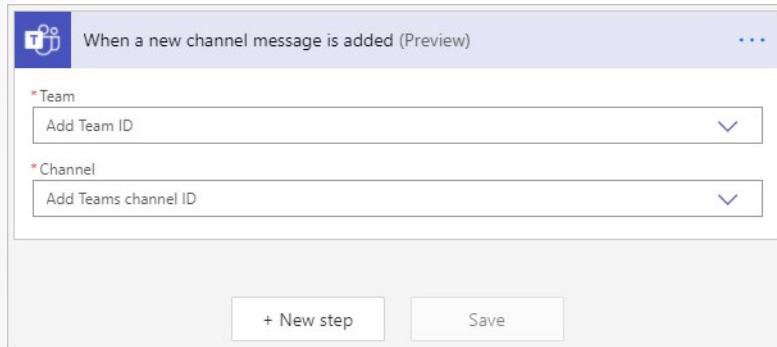
4. Select **Cloud flows** on the left to view existing flows. To create a new automated flow, select **+ New > Cloud flow > Automated**.
5. Enter a name for the flow under **Flow name**.
6. Under **Choose your flow's trigger**, scroll or search for **When a new channel message is added**. Select the trigger and then select **Create** to continue.



At this point, you have created a new flow, gave it a name, and defined the automated trigger that will start the workflow. The flow is now open in the Power Automate editor and ready to build.

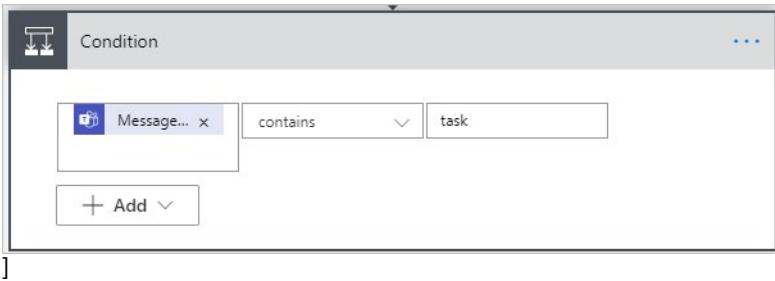
## Step 2: Build the flow in Power Automate editor

To have the flow monitor the correct Teams channel for new messages, select the **Team** and then the **Channel** from the drop-down menu selections. Select **+ New step** to continue.



Search the subject of the messages to see if they have the word "task" in them. In the search bar, enter **Condition** and then select it from the search results. Select **Choose a value** to view the available dynamic content. Dynamic content is the information that the flow has available based on the triggers and previous steps that have taken place.

1. From the list of dynamic content, select **Message subject**.
2. In the middle drop-down menu, select **contains**.
3. Enter **task** in the box on the right.

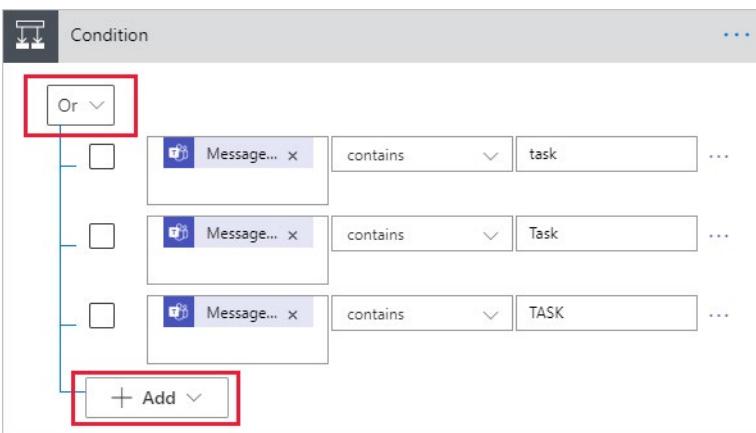


Now, when the **Message subject** contains the word "task," it will perform actions in the **If yes** area. The conditions in Power Automate are case-sensitive, so you'll need to add a few more conditions to detect common variations such as "Task" and "TASK." Select **+ Add** and then select **+ Add Row**.

A new selection will appear at the top of the **Condition** window to switch between the following statements:

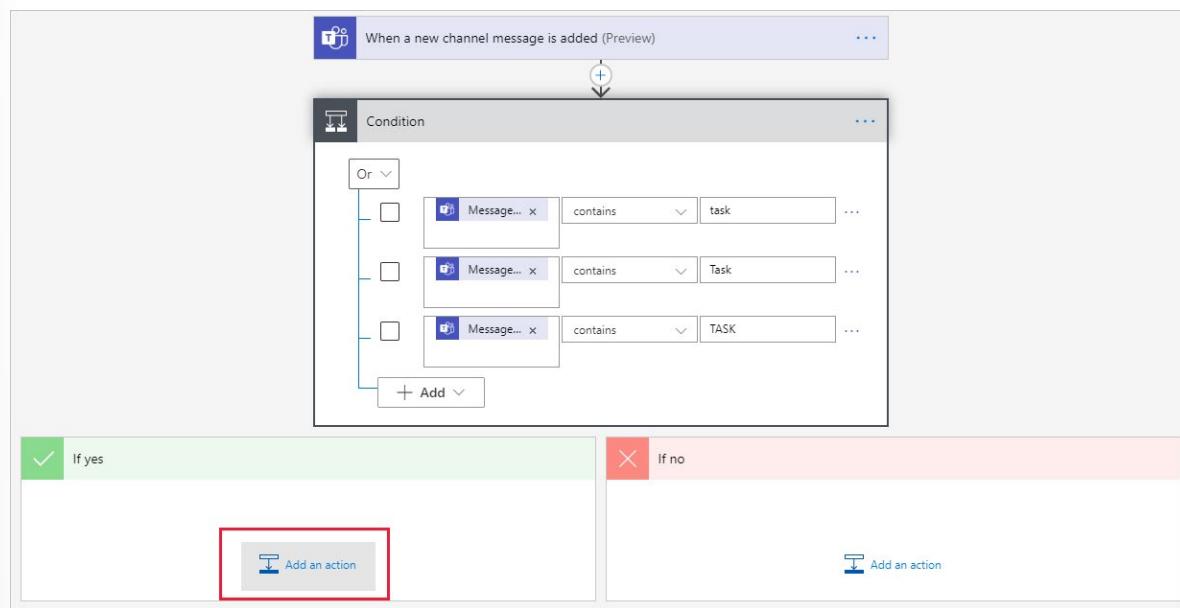
- **And** - All conditions must be true.
- **Or** - Only one of the conditions must be true.

Change the selection to **Or** and then fill out the other conditions for detecting "**Task**" and "**Tasks**". For each new row, use **Message subject** from dynamic content and set the condition to **contains**. Select **+ Add** to add more rows.

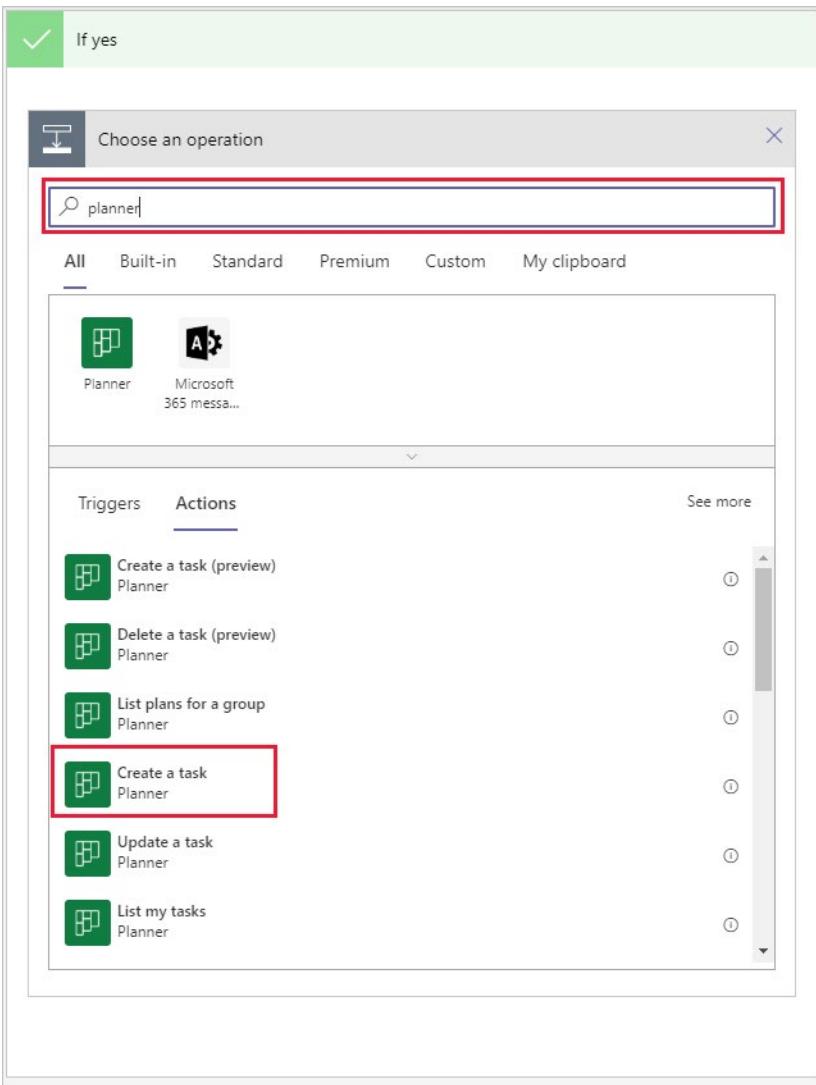


The condition action provides two other areas. If the condition is true, then you will want a new Planner task created. No actions are needed if the condition is false, so nothing more is needed in the **If no** condition.

For the **If yes** condition box, select **Add an action**.



Enter **planner** and then select **Create a task** from the results.

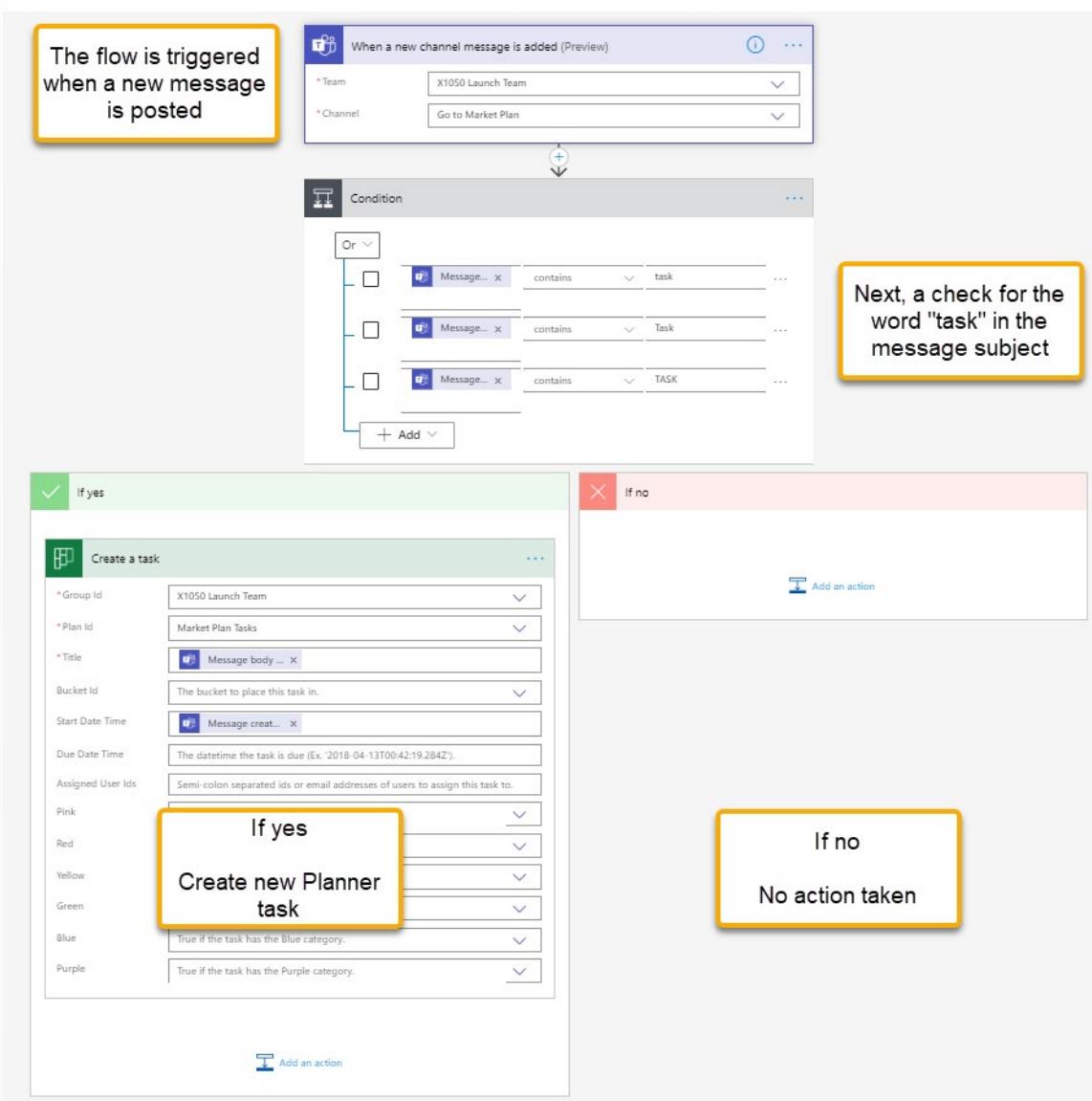


It might take up to a minute for Power Automate to authenticate into Planner. When Planner comes up, fill out the needed information for the Planner task.

| Field           | Value                                                   |
|-----------------|---------------------------------------------------------|
| Group ID        | Select the group that has your Planner                  |
| Plan ID         | Select the Planner for the new task                     |
| Title           | Select dynamic content <b>Message body content</b>      |
| Start Date Time | Select dynamic content <b>Message created Date-Time</b> |

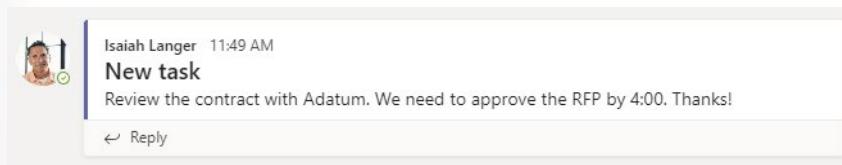
When you have completed the task, select **Save** at the bottom of the editing window or in the toolbar to complete the flow.

You can now review the flow and see how it works. The flow is triggered when a new message is posted in a specific Teams channel. Next, you should check for the word "task" in the message subject. If yes, the system creates a new Planner task. If no, the system won't take action.

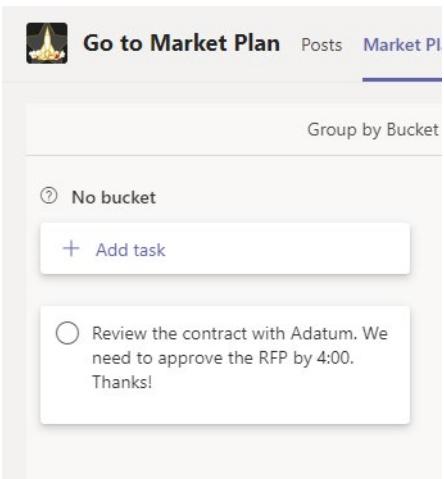


### Step 3: Test the flow

Send a message to the Teams channel that the flow is monitoring. In the subject line, enter **task**.



The Teams connector checks for a new message in three-minute intervals. Next, open Planner and watch for the new task to be created. The message body should be the task title. Select the task to check the start date and see if it was set to the same date as the message.



You can view the flow's run history for the last 28 days to see how long it takes the flow to run and if it was successful:

1. Open Power Apps for Teams.
2. From the **Build** tab, select the team that has the flow and then select **See all**.
3. Select **Cloud flows** and then select the flow to view more information and the run history.

The screenshot shows the Microsoft Power Apps interface within the Microsoft Teams application. The left sidebar includes icons for Activity, Chat, Teams, Calendar, Calls, Files, and Power Apps (which is selected). The main area displays a flow named "Create Planner Task from Teams" with the following details:

| Details |                                | Edit     |                 |
|---------|--------------------------------|----------|-----------------|
| Flow    | Create Planner Task from Teams | Status   | On              |
| Owner   | Grady Archie                   | Created  | Apr 2, 12:00 PM |
|         |                                | Modified | Apr 2, 12:07 PM |
|         |                                | Type     | Automated       |
|         |                                | Plan     | Per-user plan   |

On the right, there are sections for "Connection References" (OneDrive for Business Permissions, Microsoft Teams) and "Owners" (Grady Archie). Below these, the "28-day run history" section is shown, containing the following data:

| Start                       | Durati... | Status    |
|-----------------------------|-----------|-----------|
| Apr 2, 12:08 PM (6 min ago) | 17 ms     | Succeeded |
| Apr 2, 12:05 PM (9 min ago) | 19 ms     | Succeeded |
| Apr 2, 12:05 PM (9 min ago) | 32 ms     | Succeeded |

Select the run to see more information and troubleshoot each step of the flow.

## Exercise-Add a flow to your Power Apps app

Flows can be used to automate a sequence of actions from within your Power Apps application. For example, when a new customer record is created, you could start an approval process, add a new SharePoint list item, or email a forms survey to your customer. This feature allows for more consistent business processes and less manual work.

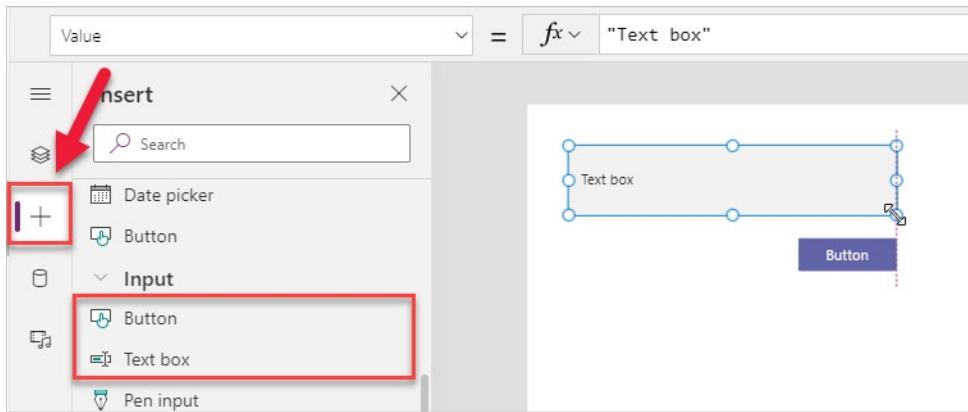
The key feature that makes this ability so powerful is that needed information is passed from the app to a flow when it's triggered. The following steps demonstrate how to connect a flow to Power Apps with a simple app to send a message to Teams. The goal is when **Button1** is pressed, the contents of **TextBox1** will be passed to the flow to become a message in Teams.

### Step 1 - Create the Power Apps application

To create the Power Apps application, follow these steps:

1. Open the Power Apps app in Teams and select **Create an app**.
2. Select a team to store the app and then select **Create**.
3. Name the app **Trigger Flow From App** and then select **Save**.

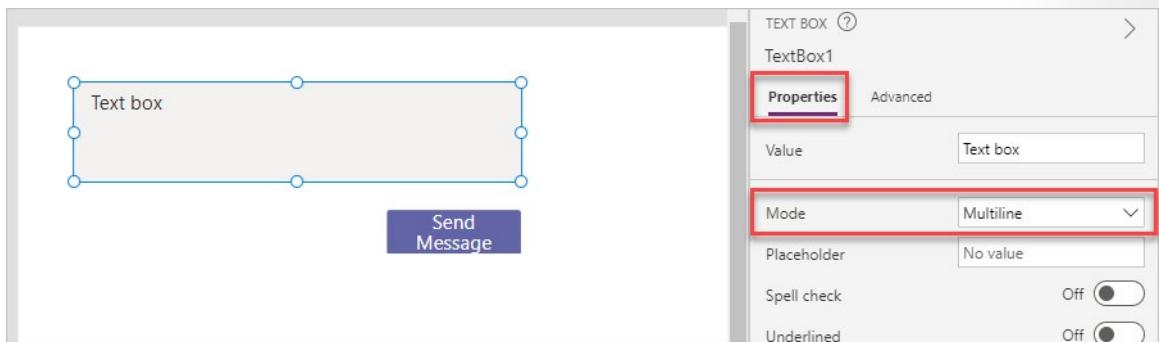
4. To open a blank screen without the hero template, select **New Screen > Blank**.
5. From the **Insert** menu, choose a **Text box** and **Button** from the **Input** section. Position the options on the screen by selecting and dragging them around. Resize the text box by first selecting it and then selecting and dragging the corners.



6. Change the following properties of **TextBox1** and **Button1** first by selecting them and using the **Properties** tab.

**TextBox1 Mode - Multiline**

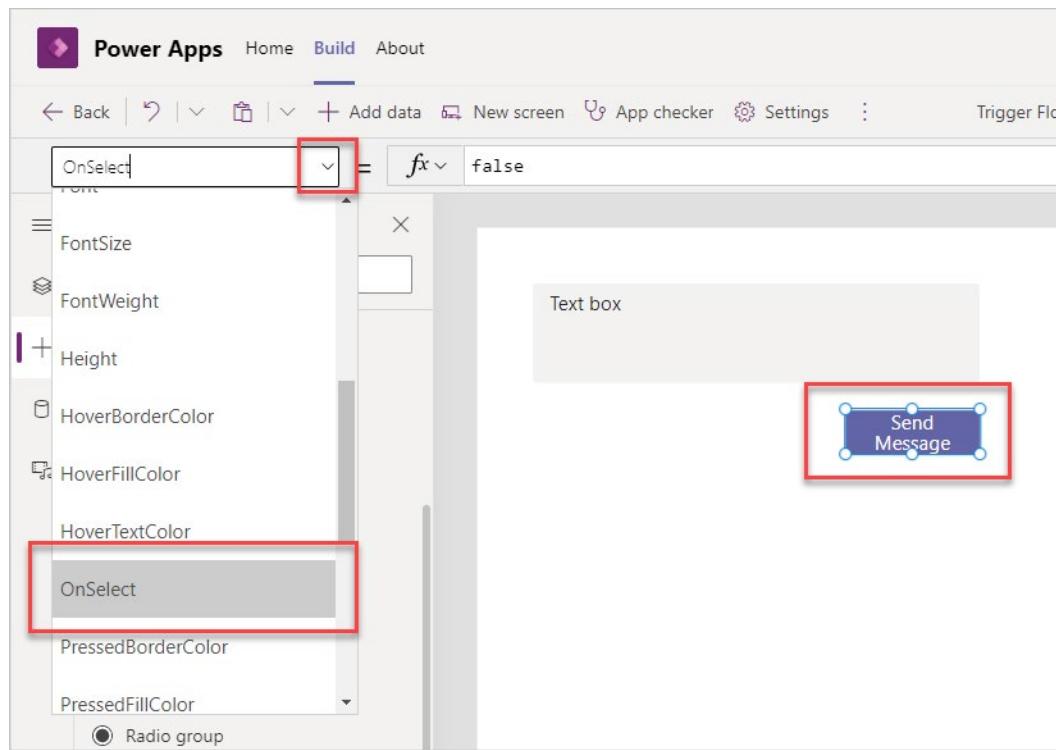
**Button1 Text - Send Message**



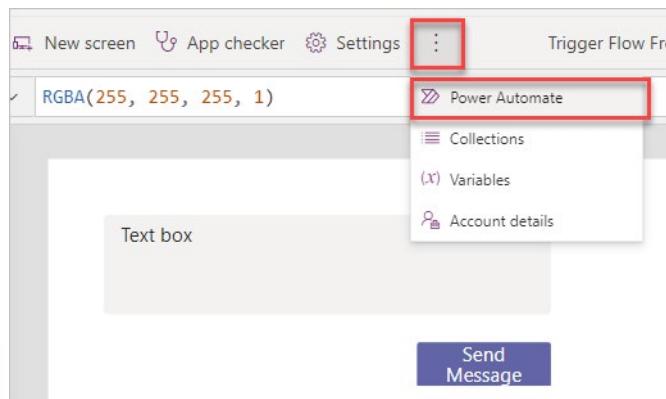
## Step 2 - Create and connect a flow

To create and connect a flow, follow these steps:

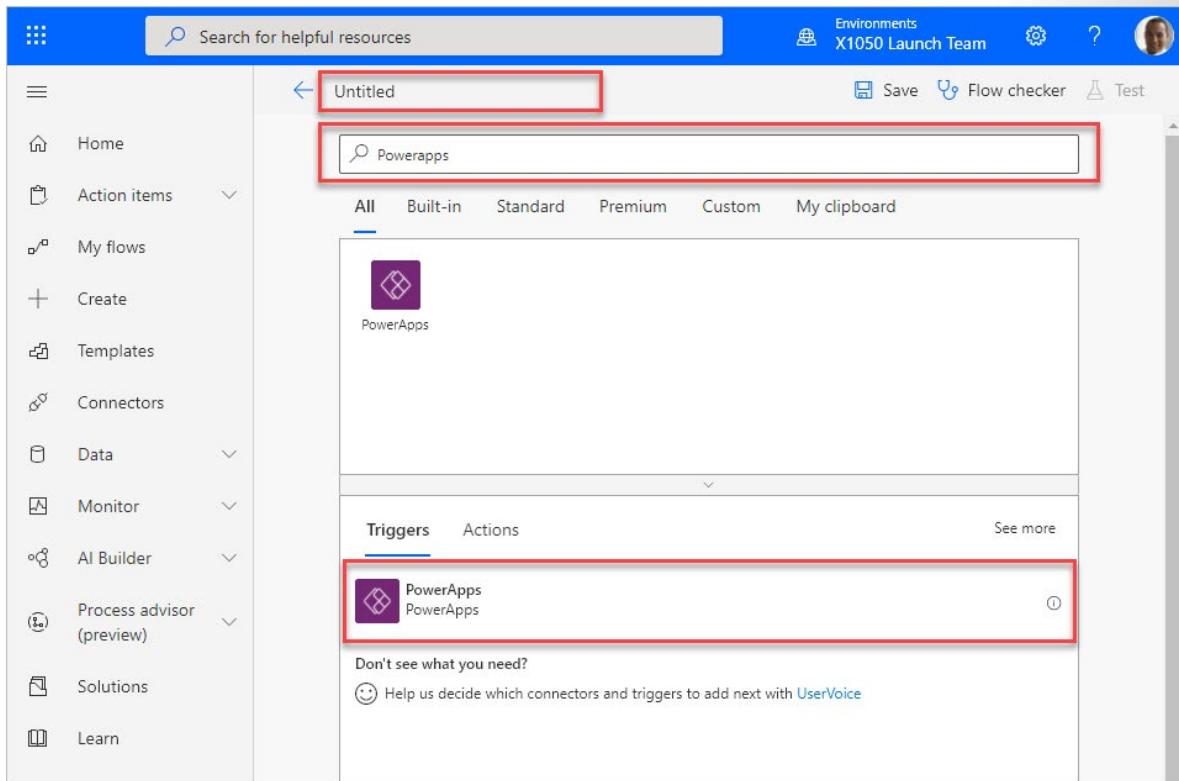
1. To connect a flow, select the item and event that will start the flow. Because you want the flow to trigger when **Button1** is pressed, select the button and then select **OnSelect** from the drop-down menu.



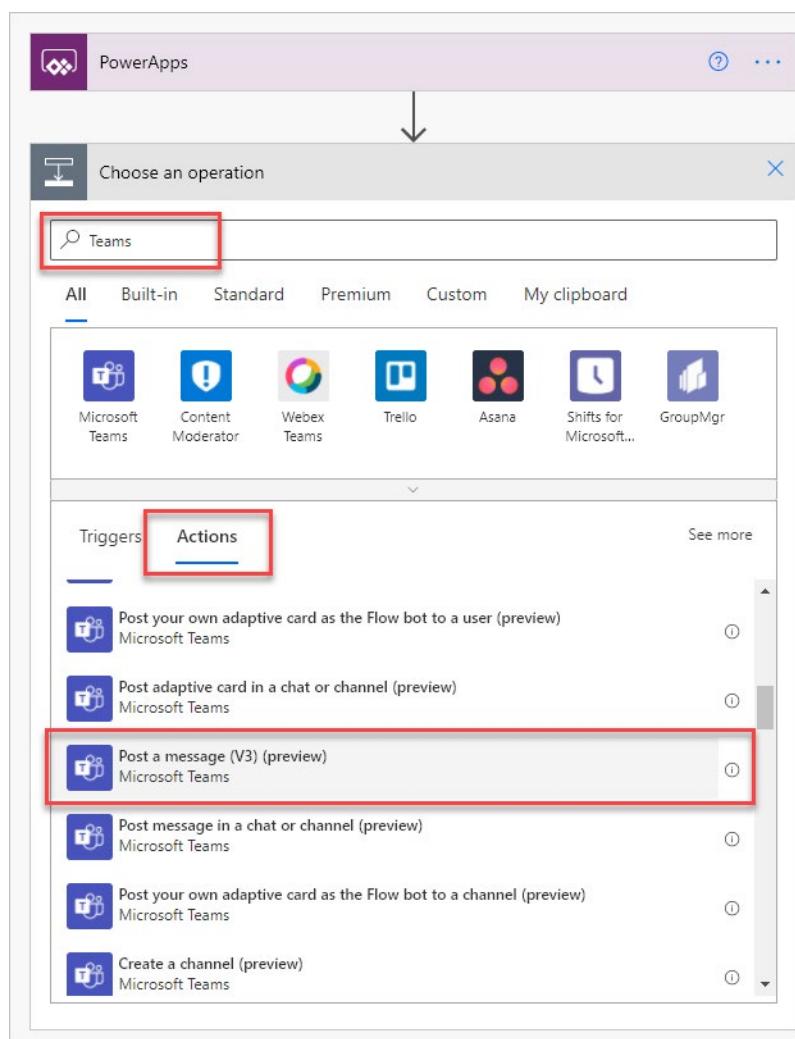
2. To create a flow and link it to the app, select the ellipsis drop-down menu and then select **Power Automate**.



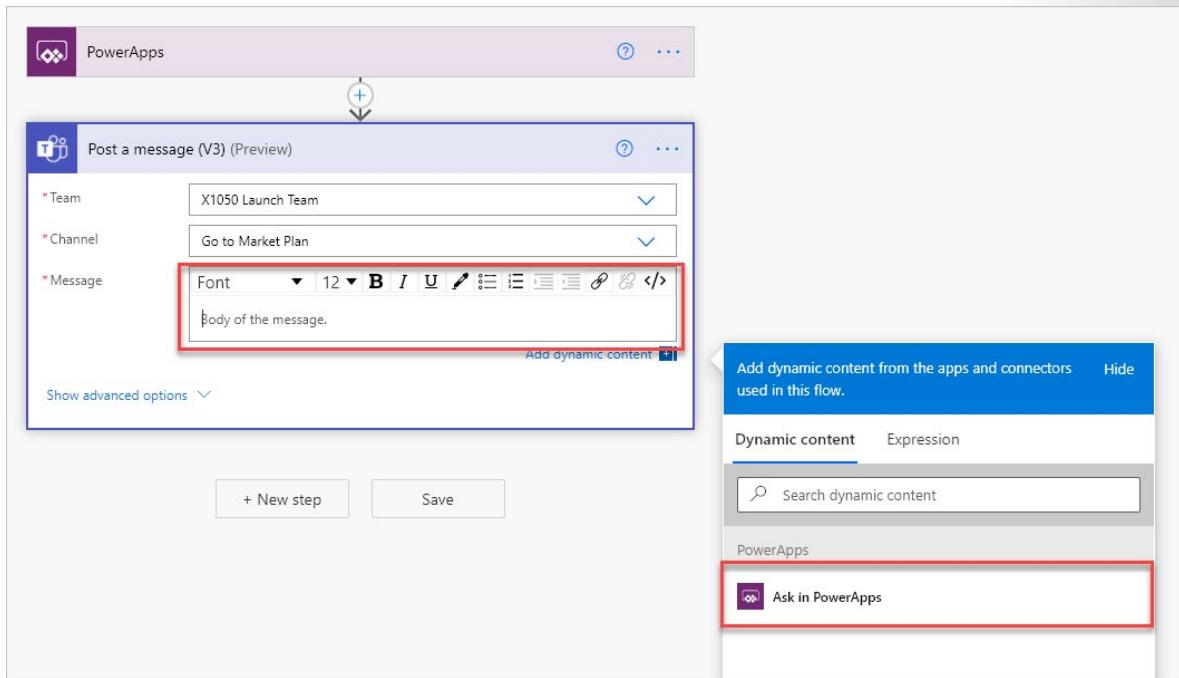
3. Select **Create a new flow** and the Power Automate editor will launch in a web browser.
4. Give your flow a new name by selecting **Untitled** and then entering **Power Apps to Teams Message**. Enter **Power Apps** in the search field and then select **Power Apps** in the **Triggers** section.



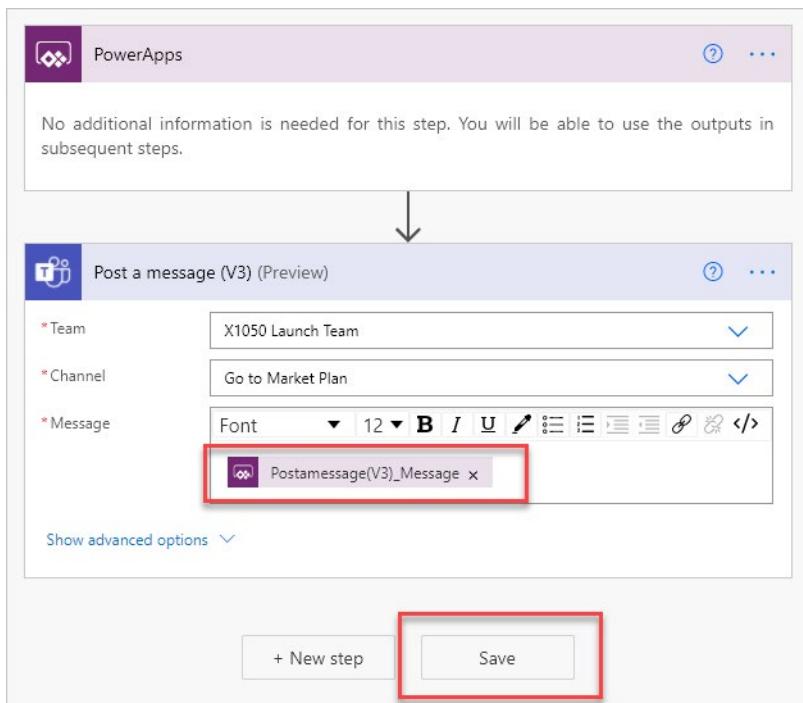
5. The flow is now editable. It only has Power Apps defined as the trigger. To add an action, select **New Step**.
6. Enter **Teams** in the search box and then scroll down under **Actions** and select **Post a message (V3) (preview)**.



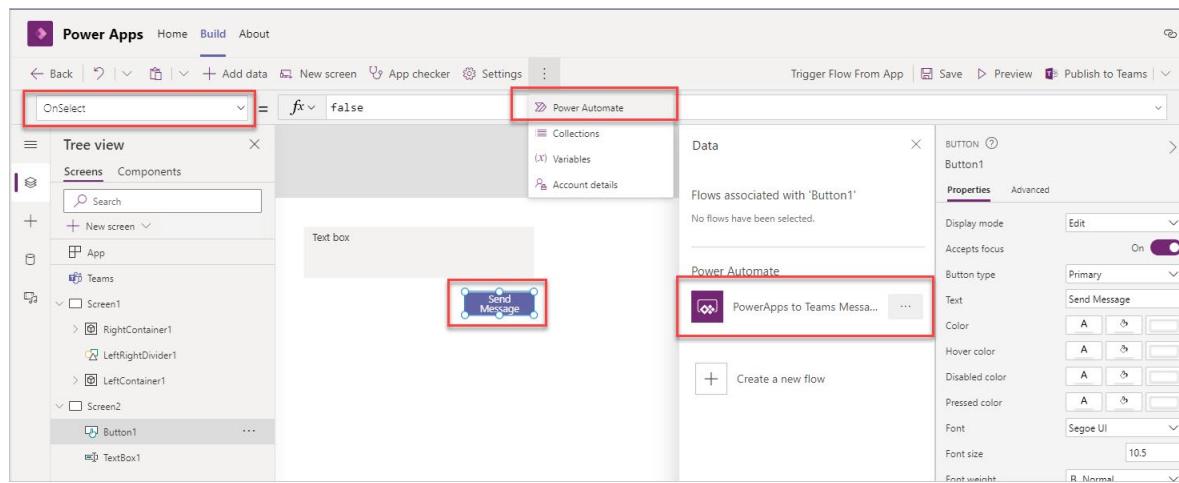
7. Select the **Team** and **Channel** that you want the message to be sent to.
8. The Team message will be provided by the Power Apps application when it's triggered. Select in the **Message** text box and then select **Ask in Power Apps** as the dynamic content.



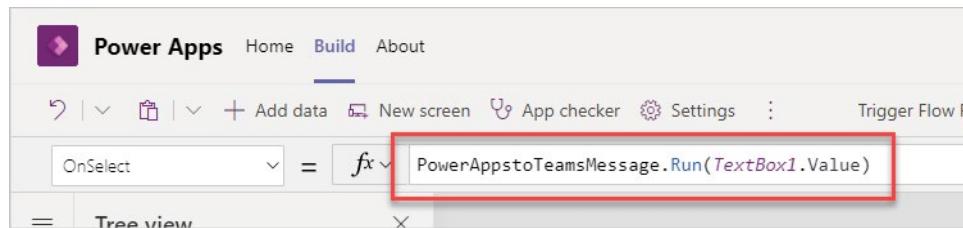
9. When **Ask in Power Apps** was selected as the dynamic content for the message, it automatically created a variable and gave it a name. Select **Save** at the end of the flow.



10. With the flow saved, return to the Power Apps editor that you started in. The new flow should appear in the Power Apps editor under the **Data** pop-out window. If, for some reason, that window is closed, select **Button1** and check that **OnSelect** is selected in the drop-down menu. Select the ellipsis menu and then select **Power Automate**. Select the flow to continue.



11. The command to run the flow will appear in the formula bar. It will be in the format of **[FlowName].Run()**. This format is only the first part of the formula. You will now need to pass in the message text for the flow. For this example, enter **TextBox1.Value** at the end of the formula.



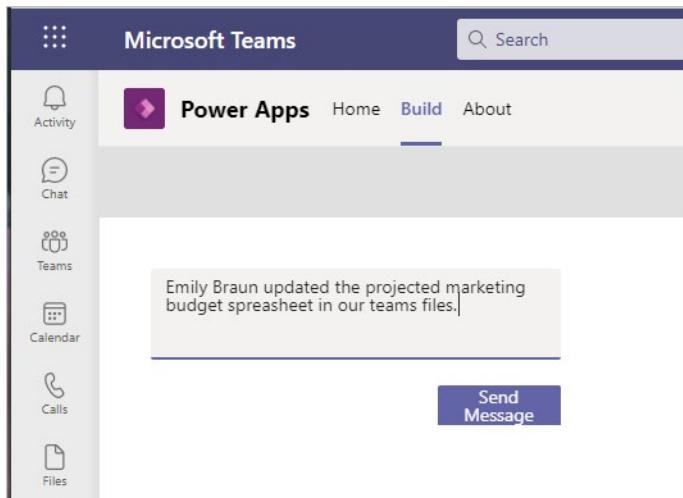
To learn more about formulas in Power Apps, see the links in the Summary unit at the end of this module.

12. Select **Save** in the toolbar.

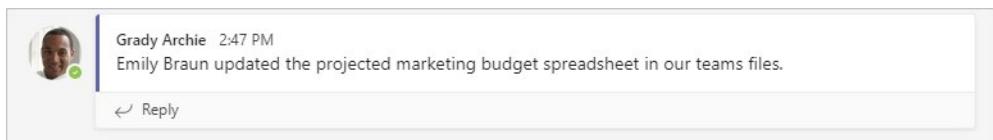
### Step 3 - Test the app and flow

Before publishing any application to **Teams**, you can test it by selecting **Preview** from the toolbar.

1. From the toolbar, select **Preview**.
2. Enter a message into the text box and then select **Send Message**.



3. Switch to the Teams channel to see the message in Teams.



4. In the Power Apps editor, close preview mode by selecting the **X** icon or by pressing the **esc** key. Because this application is only a test, it will not be published to Teams.

If you need to edit a flow that you have connected to Power Apps:

1. Open the Power Apps app in Teams.
2. Select the **Build** menu.
3. Select **See all**.
4. Select **Cloud flows** from the left menu and then select the flow from the list.

## Schedule a flow

Some business processes and tasks are done on a schedule. This schedule could include a weekly reminder for the team to turn in expense reports, a check for overdue tasks each day, or for sending out a daily report. With Dataverse for Teams scheduled flows, you can create scheduled workflows to automate these actions, making them more consistent and less time-consuming.

To create a new schedule flow:

1. Open Teams and select the Power Apps for Teams app.
2. From the **Build** tab, select the team on the left to store the flow.
3. Under the **Built by this team** tab, select **Select all**.
4. Select **+ New > Cloud flow > Scheduled**.
5. Provide a **Name** and select a **Starting** date and time for the flow. This action will set the first time that the flow will run.
6. For the **Repeat every** section, set the amount of time until the flow will run again by setting the number of seconds, minutes, hours, days, weeks, or months between each flow run.

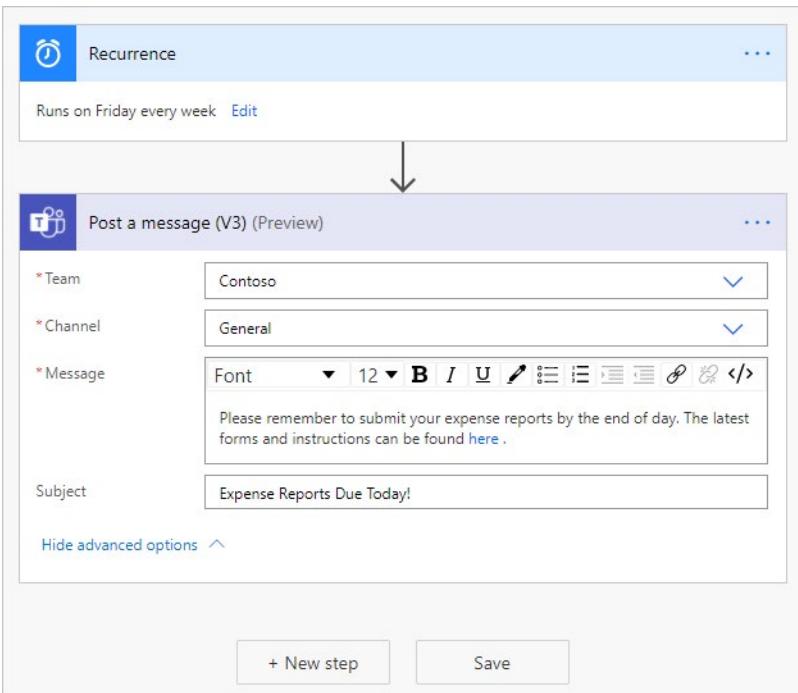
The screenshot shows the 'Build a scheduled cloud flow' interface. On the left, there's a diagram of a computer monitor with a grid icon, a blue alarm clock icon above it, and a small clock icon on the desk. Below the monitor is a keyboard. To the right is a potted plant. A text box says: 'Stay on top of what's important without the effort—you choose when and how often the flow runs.' Examples listed are: 'Automate team reminders to submit expense reports' and 'Auto-backup data to designated storage on a regular basis'. On the right, there's a form for scheduling. It includes fields for 'Flow name' (with placeholder 'Add a name or we'll generate one'), 'Run this flow \*' (with 'Starting' date '4/8/21' and time '10:00 AM'), and 'Repeat every' (set to '1 Minute'). A dropdown menu for 'This flow will run:' shows options: Month, Week, Day, Hour, Minute (which is selected), and Second. At the bottom are 'Skip', 'Create' (highlighted in blue), and 'Cancel' buttons.

Selecting **Week** will bring up more options to define which days of the week to run the flow. For example, you can run the flow on Tuesday and Thursday each week.

This screenshot shows the 'Repeat every' configuration for a weekly schedule. It displays 'Repeat every 1 Week'. Below it, under 'On these days', there are seven circular buttons labeled S, M, T, W, T, F, S. The 'T' buttons for Tuesday and Thursday are highlighted in blue, indicating they are selected. A note below says: 'This flow will run: On Tuesday, Thursday every week'.

7. Select **Create** to open the Power Automate editor and start building the rest of the flow.

After a schedule flow has been triggered, it will process the tasks that are defined in the flow just like any other flow. The following example shows a simple flow to send out a reminder to submit expense reports as a Teams message every Friday. Use the text editing controls in the message window to create hyperlinks to help make finding important resources easier for your users.



## Summary

Bring automation and workflows to Microsoft Teams by using Power Automate and Dataverse for Teams. By using automated and scheduled flows, you can build powerful workflows in Microsoft Teams and increase productivity in your Teams work. In this module, you learned how to build a flow to respond to changes in Teams, extend a Power Apps application to trigger a flow, and schedule a flow to complete routine tasks.

## Links related to modules for specific topics

For more information, see the following links:

- **Overview of Adaptive Cards for Microsoft Teams<sup>37</sup>**
- **Get started with canvas-app formulas in Power Apps<sup>38</sup>**

<sup>37</sup> <https://docs.microsoft.com/en-us/power-automate/overview-adaptive-cards/?azure-portal=true>

<sup>38</sup> <https://docs.microsoft.com/en-us/powerapps/maker/canvas-apps/working-with-formulas/?azure-portal=true>



## Module 5 Create and use analytics reports with Power BI

### Get started with Power BI

#### Introduction

**Microsoft Power BI** is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Whether your data is a simple Microsoft Excel workbook, or a collection of cloud-based and on-premises hybrid data warehouses, **Power BI** lets you easily connect to your data sources, visualize (or discover) what's important, and share that with anyone or everyone you want.

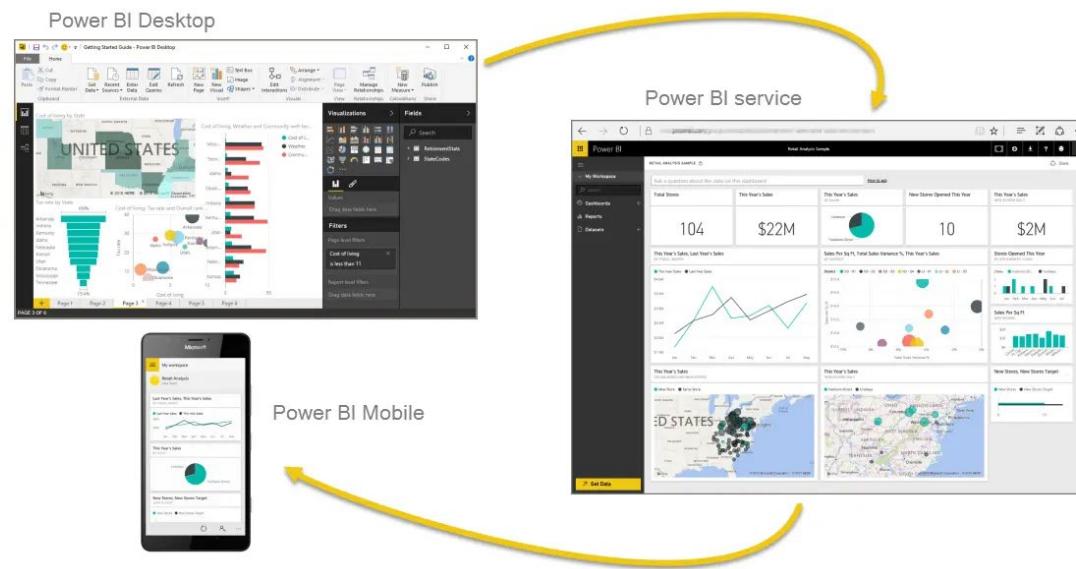


**Power BI** can be simple and fast, capable of creating quick insights from an Excel workbook or a local database. But **Power BI** is also robust and enterprise-grade, ready not only for extensive modeling and real-time analytics, but also for custom development. Therefore, it can be your personal report and visualization tool, but can also serve as the analytics and decision engine behind group projects, divisions, or entire corporations.

If you're a **beginner** with Power BI, this module will get you going. If you're a Power BI **veteran**, this module will tie concepts together and fill in the gaps.

## The parts of Power BI

Power BI consists of a Microsoft Windows desktop application called **Power BI Desktop**, an online SaaS (*Software as a Service*) service called the **Power BI service**, and mobile Power BI **apps** that are available on any device, with native mobile BI apps for Windows, iOS, and Android.



These three elements—**Desktop**, the **service**, and **Mobile** apps—are designed to let people create, share, and consume business insights in the way that serves them, or their role, most effectively.

## How Power BI matches your role

How you use Power BI might depend on your role on a project or a team. And other people, in other roles, might use Power BI differently, which is just fine.

For example, you might view reports and dashboards in the **Power BI service**, and that might be all you do with Power BI. But your number-crunching, business-report-creating coworker might make extensive use of **Power BI Desktop** (and publish Power BI Desktop reports to the Power BI service, which you then use to view them). And another coworker, in sales, might mainly use her Power BI phone app to monitor progress on her sales quotas and drill into new sales lead details.

You also might use each element of **Power BI** at different times, depending on what you're trying to achieve, or what your role is for a given project or effort.

Perhaps you view inventory and manufacturing progress in a real-time dashboard in the service, and also use **Power BI Desktop** to create reports for your own team about customer engagement statistics. How you use Power BI can depend on which feature or service of Power BI is the best tool for your situation. But each part of Power BI is available to you, which is why it's so flexible and compelling.

We discuss these three elements—**Desktop**, the **service**, and **Mobile** apps—in more detail later. In upcoming units and modules, we'll also create reports in Power BI Desktop, share them in the service, and eventually drill into them on our mobile device.

## Download Power BI Desktop

You can download Power BI Desktop from the web or as an app from the Microsoft Store on the Windows tab.

| Download Strategy | Link                                                                                                                                 | Notes                             |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| Windows Store App | <b>Windows Store</b> ( <a href="https://aka.ms/pbidesktopstore">https://aka.ms/pbidesktopstore</a> )                                 | Will automatically stay updated   |
| Download from web | <b>Download .msi</b> ( <a href="https://go.microsoft.com/fwlink/?LinkId=521662">https://go.microsoft.com/fwlink/?LinkId=521662</a> ) | Must manually update periodically |

## Sign in to Power BI service

Before you can sign in to Power BI, you'll need an account. To get a free trial, go to [app.powerbi.com](http://app.powerbi.com)<sup>1</sup> and sign up with your email address.

For detailed steps on setting up an account, see [Sign in to Power BI service](#)<sup>2</sup>

## The flow of work in Power BI

A common flow of work in Power BI begins in **Power BI Desktop**, where a report is created. That report is then published to the **Power BI service** and finally shared, so that users of **Power BI Mobile** apps can consume the information.

It doesn't always happen that way, and that's okay. But we'll use that flow to help you learn the different parts of Power BI and how they complement each other.

Okay, now that we have an overview of this module, what Power BI is, and its three main elements, let's take a look at what it's like to use **Power BI**.

## Use Power BI

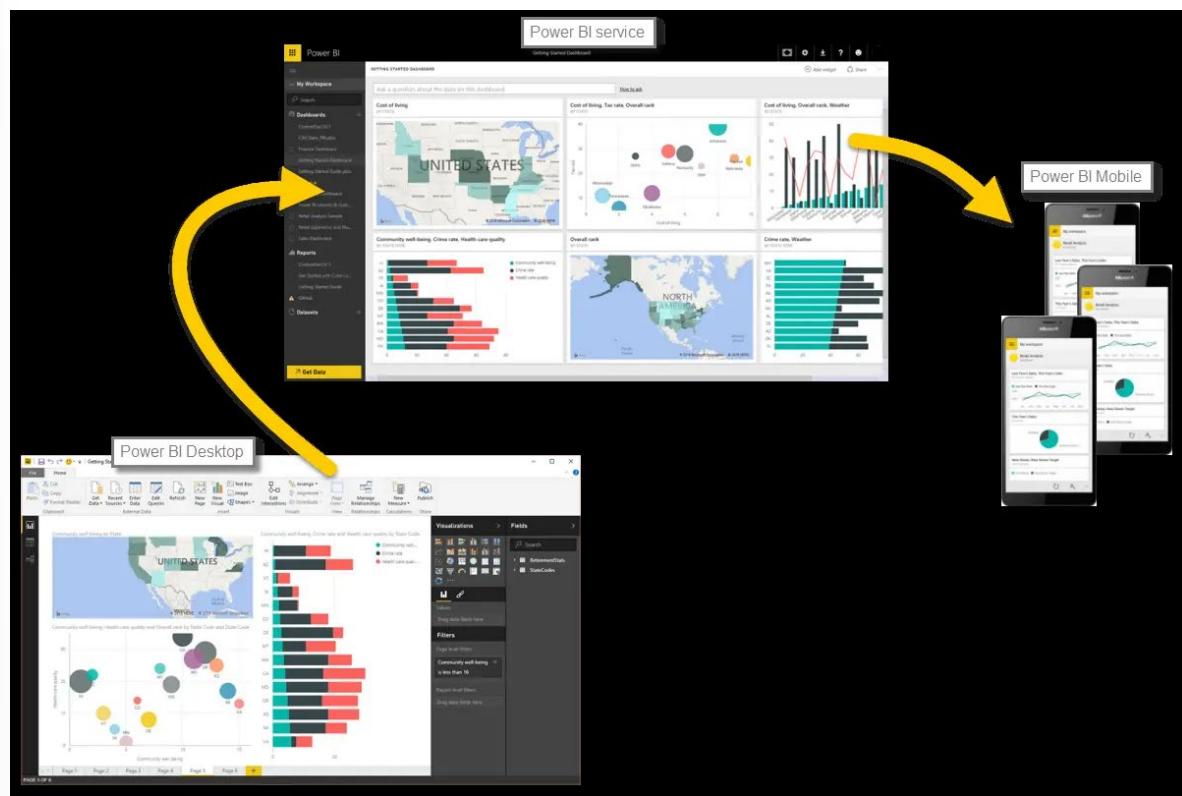
Now that we've introduced the basics of Microsoft Power BI, let's jump into some hands-on experiences and a guided tour.

The activities and analyses that you'll learn with Power BI generally follow a common flow. The **common flow** of activity looks like this:

1. Bring data into Power BI Desktop, and create a report.
2. Publish to the Power BI service, where you can create new visualizations or build dashboards.
3. Share dashboards with others, especially people who are on the go.
4. View and interact with shared dashboards and reports in Power BI Mobile apps.

<sup>1</sup> <https://go.microsoft.com/fwlink/?LinkId=2101313>

<sup>2</sup> <https://docs.microsoft.com/power-bi/consumer/end-user-sign-in>



As mentioned earlier, you might spend all your time in the **Power BI service**, viewing visuals and reports that have been created by others. And that's fine. Someone else on your team might spend their time in **Power BI Desktop**, which is fine too. To help you understand the full continuum of Power BI and what it can do, we'll show you all of it. Then you can decide how to use it to your best advantage.

So, let's jump in and step through the experience. Your first order of business is to learn the basic building blocks of Power BI, which will provide a solid basis for turning data into cool reports and visuals.

## Building blocks of Power BI

Everything you do in Microsoft Power BI can be broken down into a few basic **building blocks**. After you understand these building blocks, you can expand on each of them and begin creating elaborate and complex reports. After all, even seemingly complex things are built from basic building blocks. For example, buildings are created with wood, steel, concrete and glass, and cars are made from metal, fabric, and rubber. Of course, buildings and cars can also be basic or elaborate, depending on how those basic building blocks are arranged.

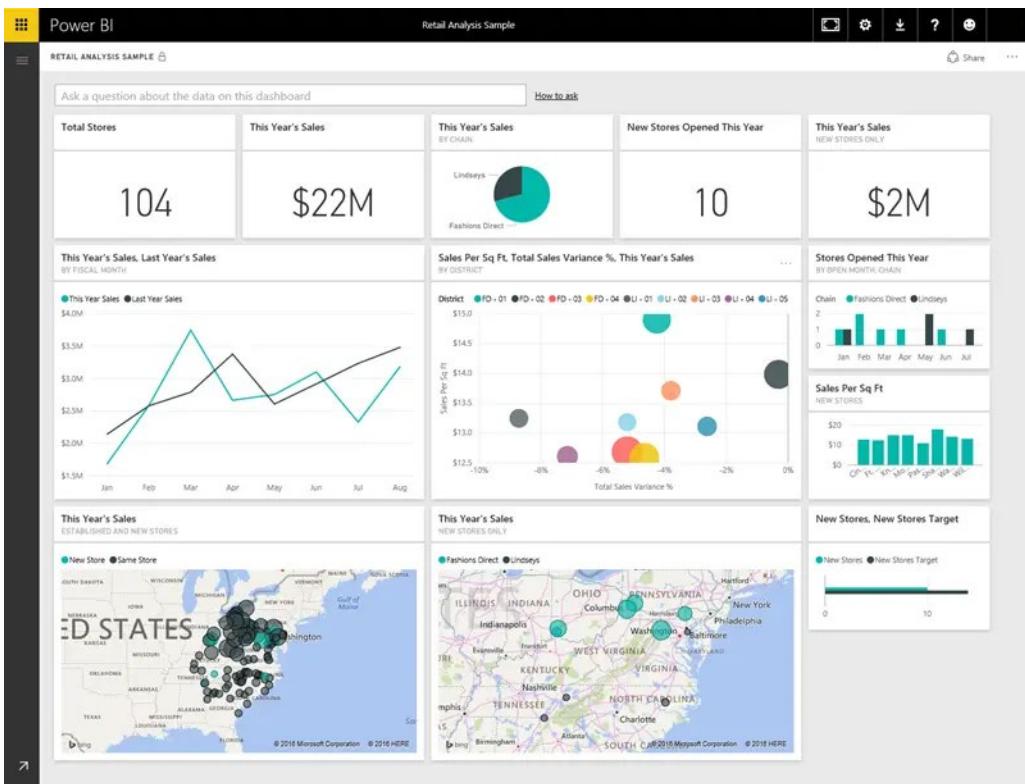
Let's take a look at these basic building blocks, discuss some simple things that can be built with them, and then get a glimpse into how complex things can also be created.

Here are the basic building blocks in Power BI:

- Visualizations
- Datasets
- Reports
- Dashboards
- Tiles

## Visualizations

A **visualization** (sometimes also referred to as a **visual**) is a visual representation of data, like a chart, a color-coded map, or other interesting things you can create to represent your data visually. Power BI has all sorts of visualization types, and more are coming all the time. The following image shows a collection of different visualizations that were created in Power BI.



Visualizations can be simple, like a single number that represents something significant, or they can be visually complex, like a gradient-colored map that shows voter sentiment about a certain social issue or concern. The goal of a visual is to present data in a way that provides context and insights, both of which would probably be difficult to discern from a raw table of numbers or text.

## Datasets

A **dataset** is a collection of data that Power BI uses to create its visualizations.

You can have a simple dataset that's based on a single table from a Microsoft Excel workbook, similar to what's shown in the following image.

|      | B    | C     | D          | E              | F      | G              | H                   |
|------|------|-------|------------|----------------|--------|----------------|---------------------|
| 1    | Year | Month | Month Name | Calendar Month | Births | Births Per Day | Births (Normalized) |
| 2119 | 2004 | 1     | January    | 1/1/2004       | 2,937  | 94.7           | 2842                |
| 2120 | 2004 | 2     | February   | 2/1/2004       | 2,824  | 97.4           | 2921                |
| 2121 | 2004 | 3     | March      | 3/1/2004       | 3,128  | 100.9          | 3027                |
| 2122 | 2004 | 4     | April      | 4/1/2004       | 2,896  | 96.5           | 2896                |
| 2123 | 2004 | 5     | May        | 5/1/2004       | 3,008  | 97.0           | 2911                |
| 2124 | 2004 | 6     | June       | 6/1/2004       | 3,047  | 101.6          | 3047                |
| 2125 | 2004 | 7     | July       | 7/1/2004       | 2,981  | 96.2           | 2885                |
| 2126 | 2004 | 8     | August     | 8/1/2004       | 3,079  | 99.3           | 2980                |
| 2127 | 2004 | 9     | September  | 9/1/2004       | 3,219  | 107.3          | 3219                |
| 2128 | 2004 | 10    | October    | 10/1/2004      | 3,547  | 114.4          | 3433                |
| 2129 | 2004 | 11    | November   | 11/1/2004      | 3,365  | 112.2          | 3365                |
| 2130 | 2004 | 12    | December   | 12/1/2004      | 3,143  | 101.4          | 3042                |
| 2131 | 2005 | 1     | January    | 1/1/2005       | 2,921  | 94.2           | 2827                |
| 2132 | 2005 | 2     | February   | 2/1/2005       | 2,699  | 96.4           | 2892                |
| 2133 | 2005 | 3     | March      | 3/1/2005       | 3,024  | 97.5           | 2926                |
| 2134 | 2005 | 4     | April      | 4/1/2005       | 3,037  | 101.2          | 3037                |
| 2135 | 2005 | 5     | May        | 5/1/2005       | 3,231  | 104.2          | 3127                |
| 2136 | 2005 | 6     | June       | 6/1/2005       | 3,163  | 105.4          | 3163                |
| 2137 | 2005 | 7     | July       | 7/1/2005       | 3,119  | 100.6          | 3018                |
| 2138 | 2005 | 8     | August     | 8/1/2005       | 3,156  | 101.8          | 3054                |
| 2139 | 2005 | 9     | September  | 9/1/2005       | 3,439  | 114.6          | 3439                |

**Datasets** can also be a combination of many different sources, which you can filter and combine to provide a unique collection of data (a dataset) for use in Power BI.

For example, you can create a dataset from three database fields, one website table, an Excel table, and online results of an email marketing campaign. That unique combination is still considered a single **dataset**, even though it was pulled together from many different sources.

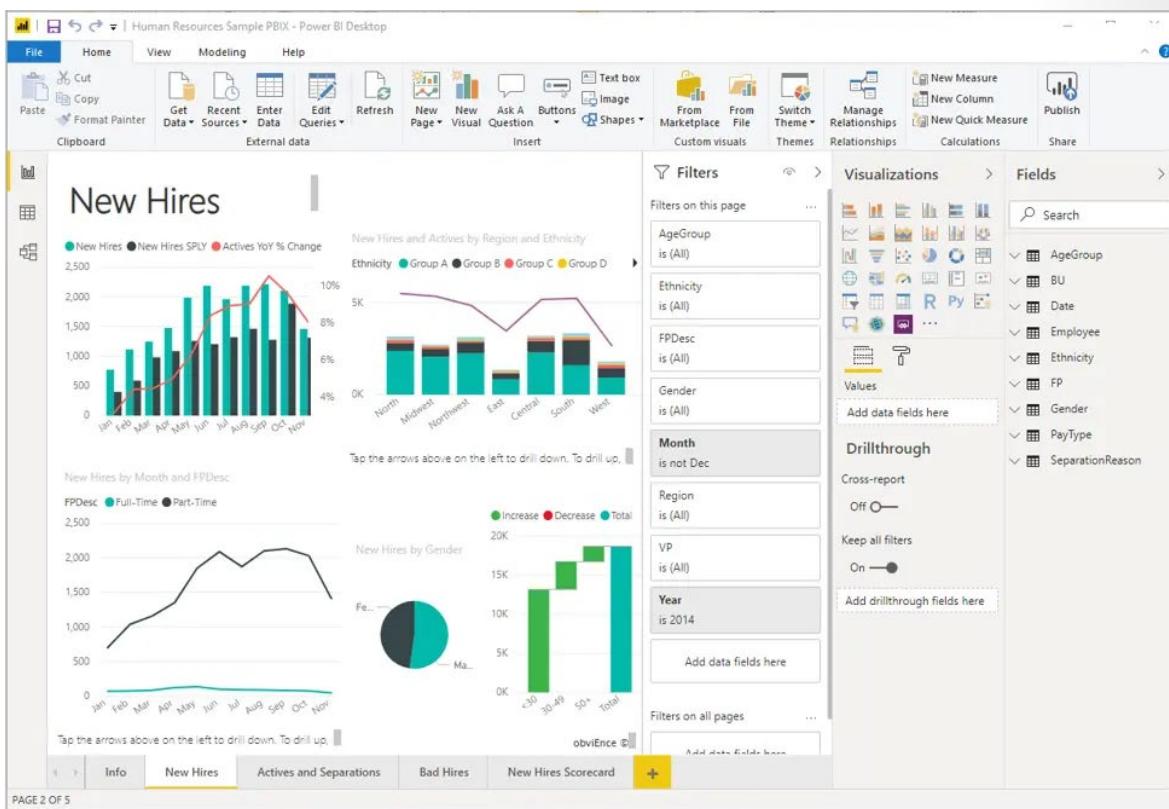
Filtering data before bringing it into Power BI lets you focus on the data that matters to you. For example, you can filter your contact database so that only customers who received emails from the marketing campaign are included in the dataset. You can then create visuals based on that subset (the filtered collection) of customers who were included in the campaign. Filtering helps you focus your data—and your efforts.

An important and enabling part of Power BI is the multitude of data **connectors** that are included. Whether the data you want is in Excel or a Microsoft SQL Server database, in Azure or Oracle, or in a service like Facebook, Salesforce, or MailChimp, Power BI has built-in data connectors that let you easily connect to that data, filter it if necessary, and bring it into your dataset.

After you have a dataset, you can begin creating visualizations that show different portions of it in different ways, and gain insights based on what you see. That's where reports come in.

## Reports

In Power BI, a **report** is a collection of visualizations that appear together on one or more pages. Just like any other report you might create for a sales presentation or write for a school assignment, a report in Power BI is a collection of items that are related to each other. The following image shows a **report** in Power BI Desktop—in this case, it's the second page in a five-page report. You can also create reports in the Power BI service.



Reports let you create many visualizations, on multiple pages if necessary, and let you arrange those visualizations in whatever way best tells your story.

You might have a report about quarterly sales, product growth in a particular segment, or migration patterns of polar bears. Whatever your subject, reports let you gather and organize your visualizations onto one page (or more).

## Dashboards

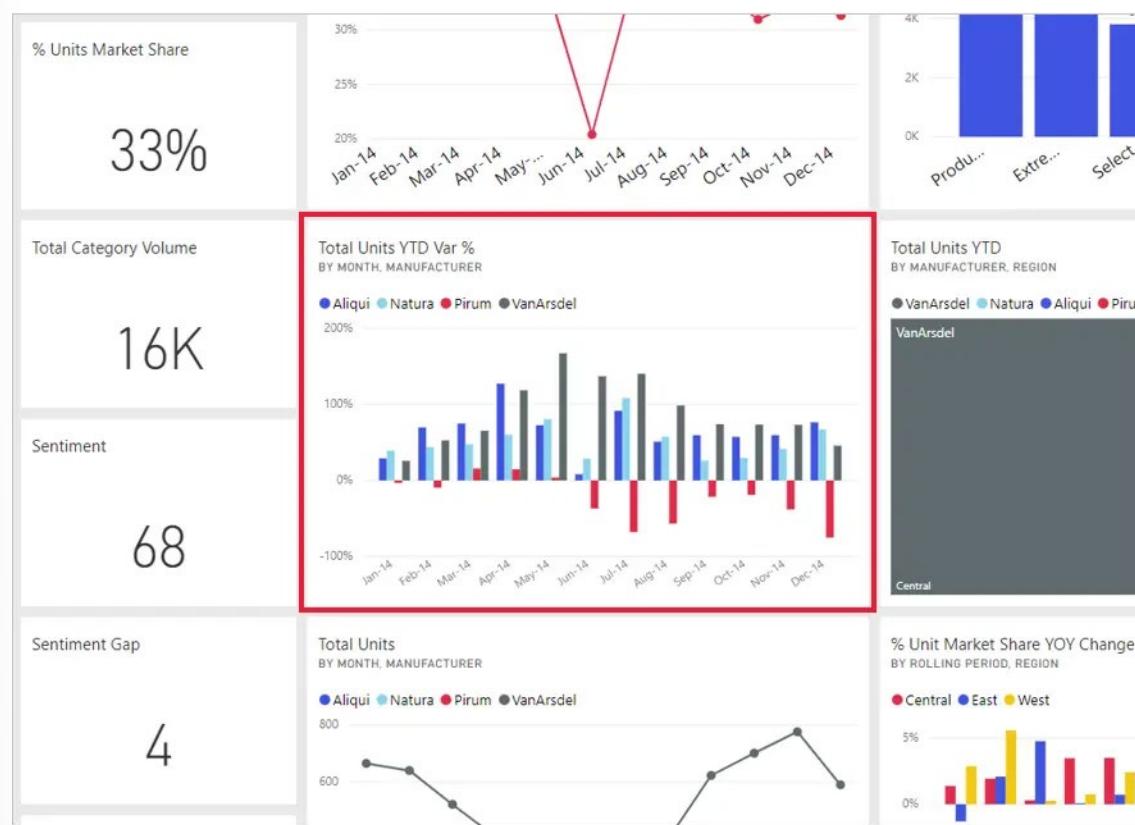
When you're ready to share a report, or a collection of visualizations, you create a **dashboard**. Much like the dashboard in a car, a Power BI **dashboard** is a collection of visualizations from a single page that you can share with others. Often, it's a selected group of visualizations that provide quick insight into the data or story you're trying to present.

A dashboard must fit on a single page, often called a canvas (the canvas is the blank backdrop in Power BI Desktop or the service, where you put visualizations). Think of it like the canvas that an artist or painter uses—a workspace where you create, combine, and rework interesting and compelling visuals.

You can share dashboards with other users or groups, who can then interact with your dashboards when they're in the Power BI service or on their mobile device.

## Tiles

In Power BI, a **tile** is a single visualization on a dashboard. It's the rectangular box that holds an individual visual. In the following image, you see one tile, which is also surrounded by other tiles.



When you're *creating* a dashboard in Power BI, you can move or arrange tiles however you want. You can make them bigger, change their height or width, and snuggle them up to other tiles.

When you're *viewing*, or *consuming*, a dashboard or report—which means you're not the creator or owner, but the report or dashboard has been shared with you—you can interact with it, but you can't change the size of the tiles or their arrangement.

## All together now

Those are the basics of Power BI and its building blocks. Let's take a moment to review.

Power BI is a collection of services, apps, and connectors that lets you connect to your data, wherever it happens to reside, filter it if necessary, and then bring it into Power BI to create compelling visualizations that you can share with others.

Now that you've learned about the handful of basic building blocks of Power BI, it should be clear that you can create datasets that make sense *to you* and create visually compelling reports that tell your story. Stories told with Power BI don't have to be complex, or complicated, to be compelling.

For some people, using a single Excel table in a dataset and then sharing a dashboard with their team will be an incredibly valuable way to use Power BI.

For others, the value of Power BI will be in using real-time Azure SQL Data Warehouse tables that combine with other databases and real-time sources to build a moment-by-moment dataset.

For both groups, the process is the same: create datasets, build compelling visuals, and share them with others. And the result is also the same for both groups: harness your ever-expanding world of data, and turn it into actionable insights.

Whether your data insights require straightforward or complex datasets, Power BI helps you get started quickly and can expand with your needs to be as complex as your world of data requires. And because Power BI is a Microsoft product, you can count on it being robust, extensible, Microsoft Office–friendly, and enterprise-ready.

Now let's see how this works. We'll start by taking a quick look at the Power BI service.

## Tour and use the Power BI service

As we learned in the previous unit, the common flow of work in Microsoft Power BI is to create a report in Power BI Desktop, publish it to the Power BI service, and then share it with others, so that they can view it in the service or on a mobile app.

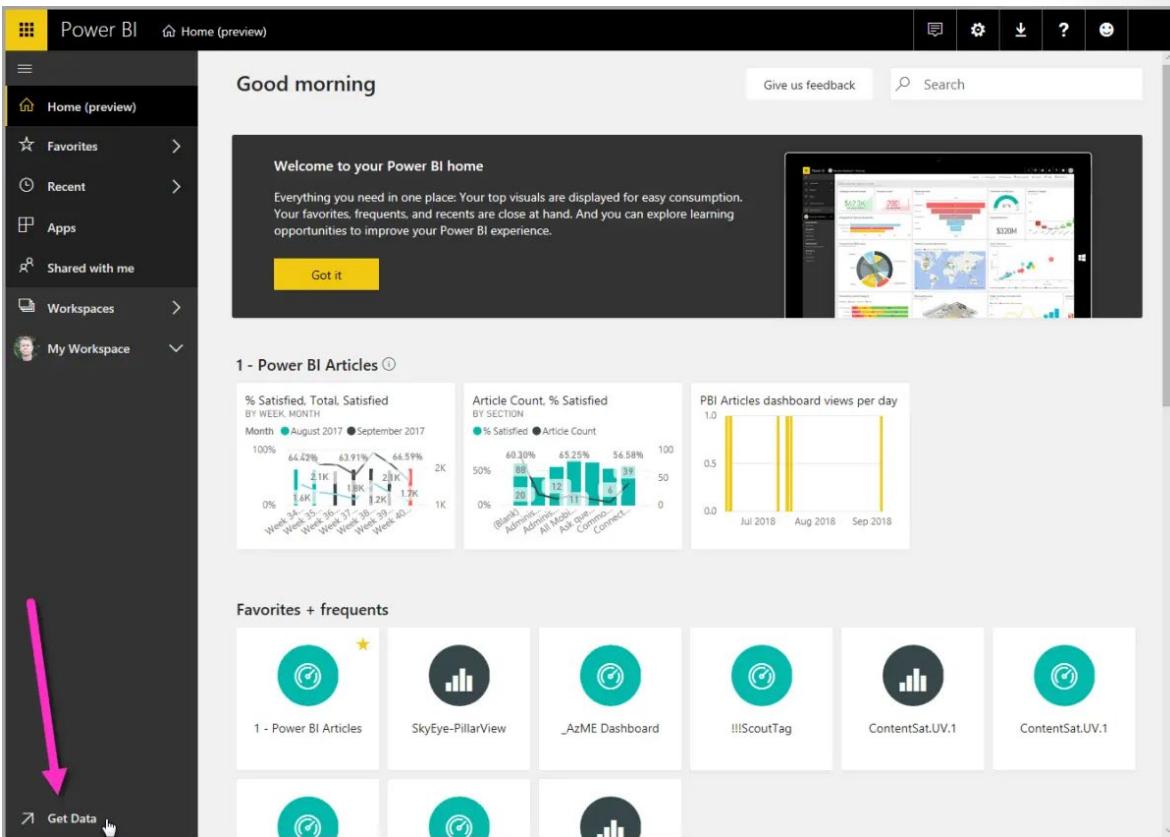
But because some people begin in the Power BI service, let's take a quick look at that first, and learn about an easy and popular way to quickly create visuals in Power BI: *apps*.

An **app** is a collection of preset, ready-made visuals and reports that are shared with an entire organization. Using an app is like microwaving a TV dinner or ordering a fast-food value meal: you just have to press a few buttons or make a few comments, and you're quickly served a collection of entrees designed to go together, all presented in a tidy, ready-to-consume package.

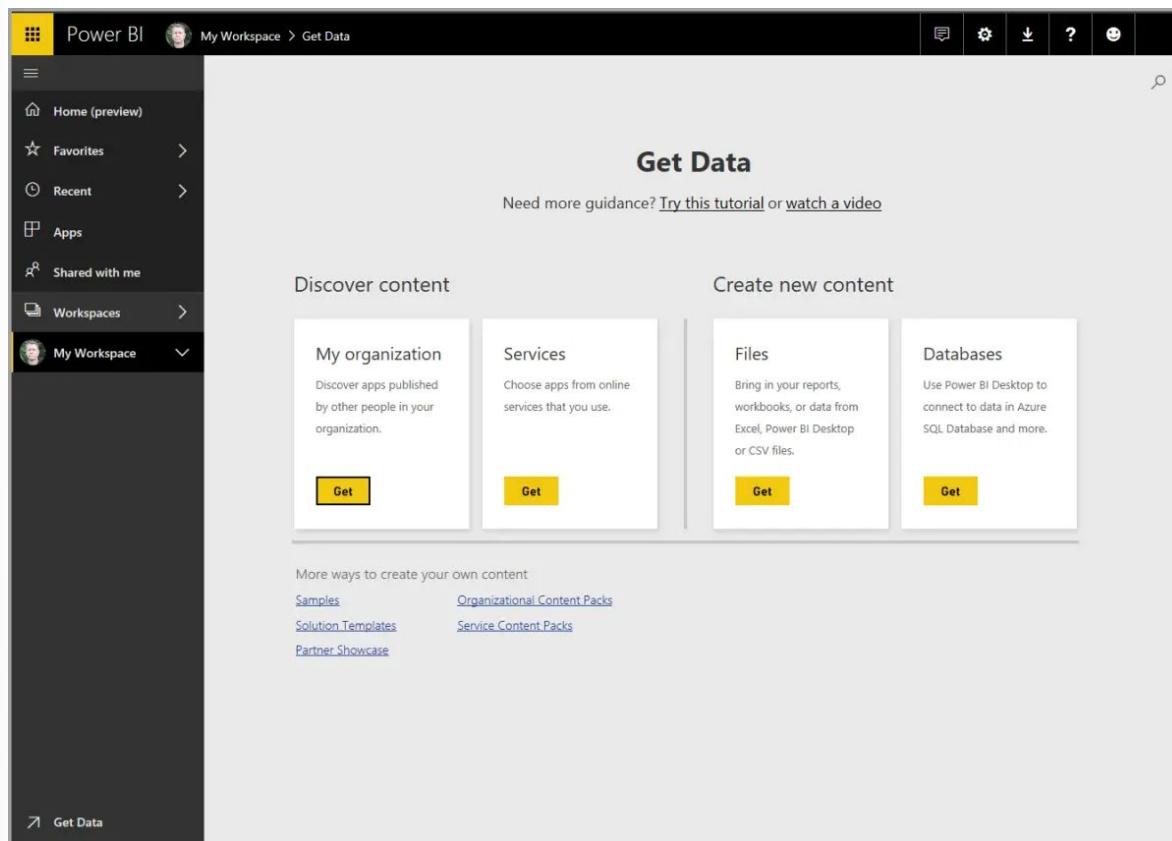
So, let's take a quick look at apps, the service, and how it works. We'll go into more detail about apps (and the service) in upcoming modules, but you can think of this as a taste to whet your appetite.

## Create out-of-box dashboards with cloud services

With Power BI, connecting to data is easy. From the Power BI service, you can just select the **Get Data** button in the lower-left corner of the home page.

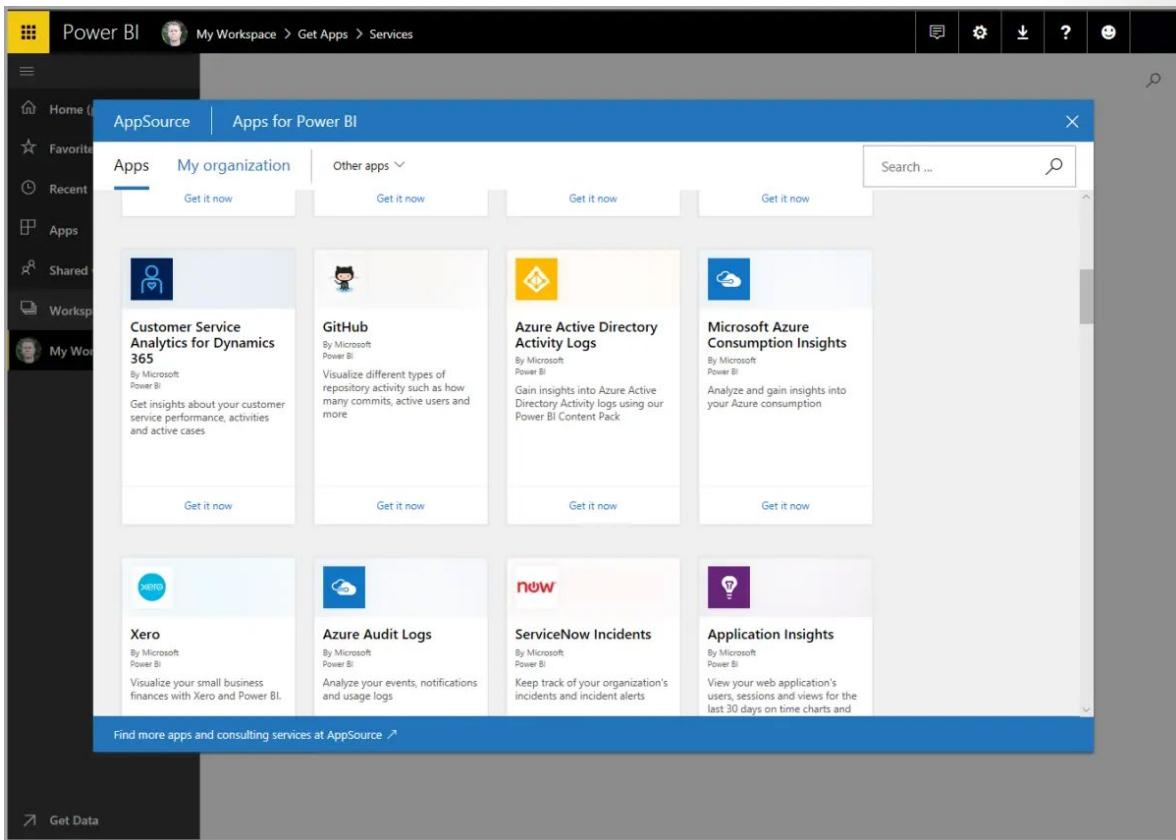


The *canvas* (the area in the center of the Power BI service) shows you the available sources of data in the Power BI service. In addition to common data sources like Microsoft Excel files, databases, or Microsoft Azure data, Power BI can just as easily connect to a whole assortment of **software services** (also called SaaS providers or cloud services): Salesforce, Facebook, Google Analytics, and more.

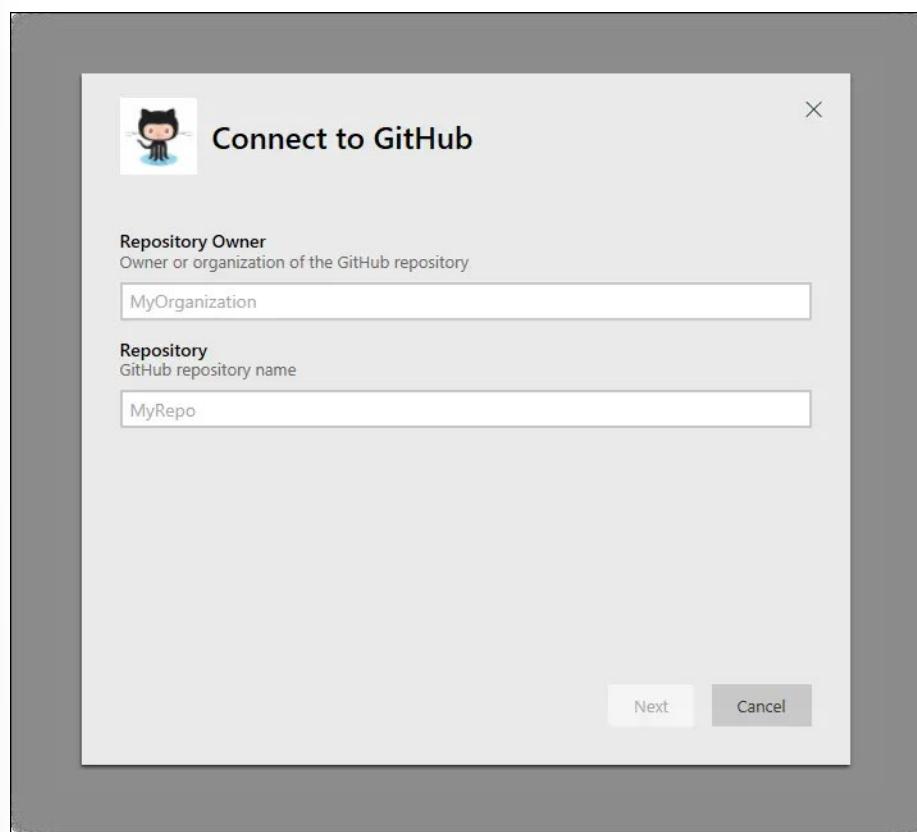


For these software services, the **Power BI service** provides a collection of ready-made visuals that are pre-arranged on dashboards and reports for your organization. This collection of visuals is called an **app**. Apps get you up and running quickly, with data and dashboards that your organization has created for you. For example, when you use the GitHub app, Power BI connects to your GitHub account (after you provide your credentials) and then populates a predefined collection of visuals and dashboards in Power BI.

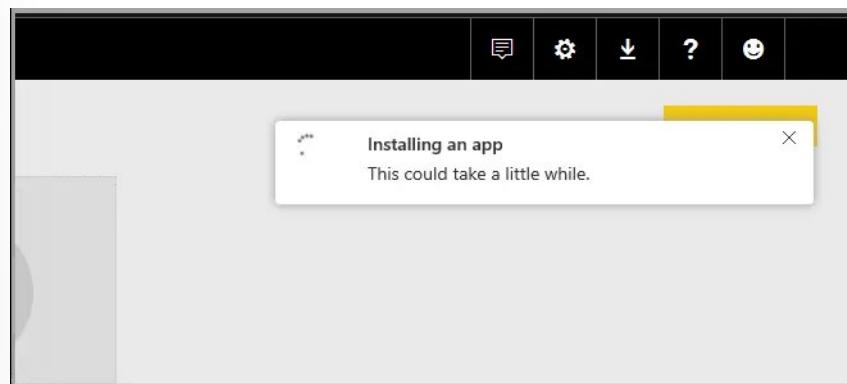
There are apps for all sorts of online services. The following image shows a page of apps that are available for different online services, in alphabetical order. This page is shown when you select the **Get** button in the **Services** box (shown in the previous image). As you can see from the following image, there are many apps to choose from.



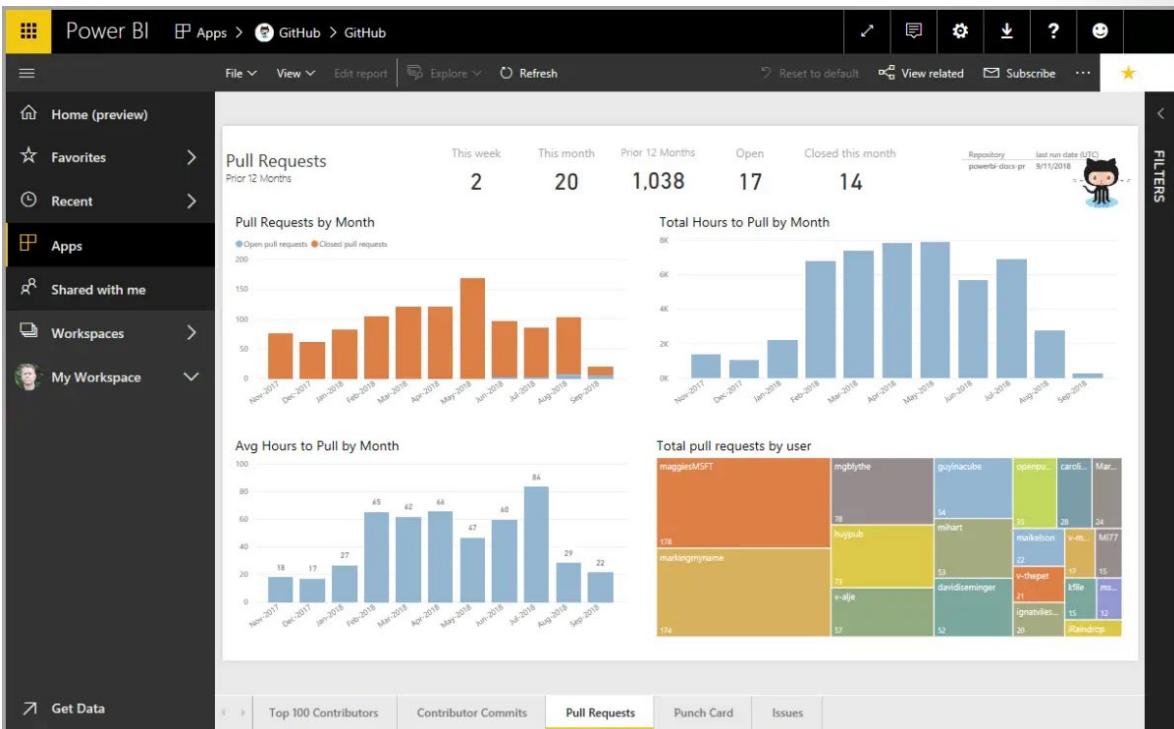
For our purposes, we'll choose **GitHub**. GitHub is an application for online source control. When you select the **Get it now** button in the box for the GitHub app, the **Connect to GitHub** dialog box appears. Note that GitHub does not support Internet Explorer, so make sure you are working in another browser.



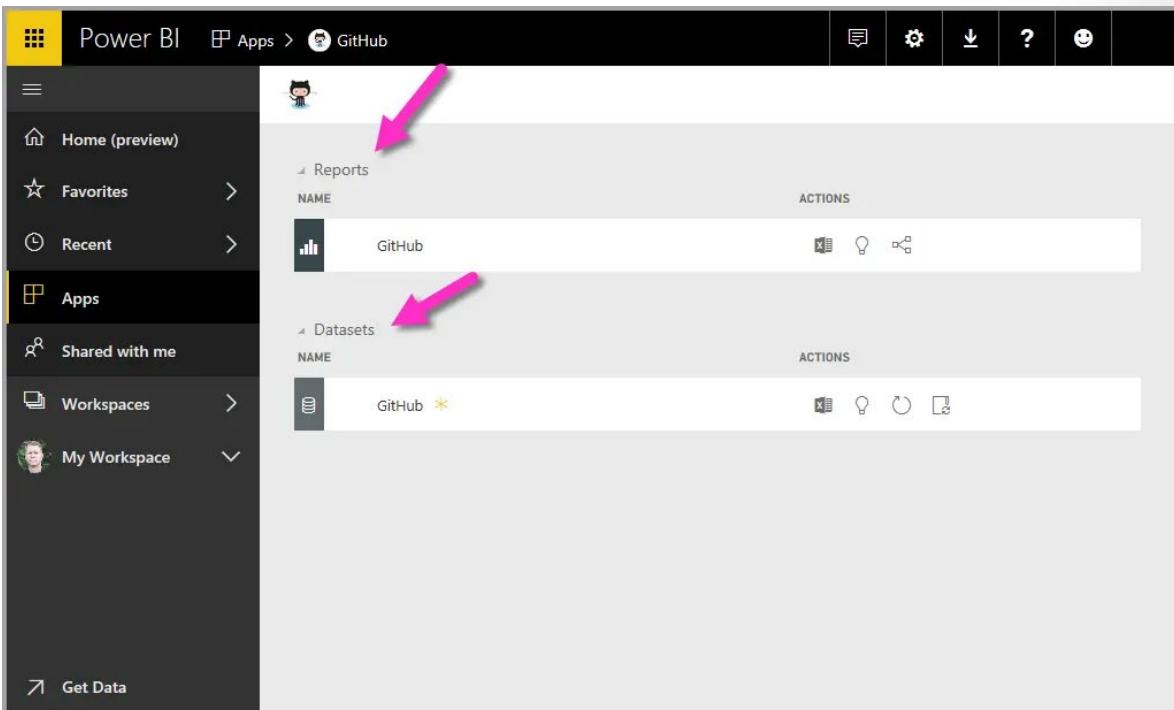
After you enter the information and credentials for the GitHub app, installation of the app begins.



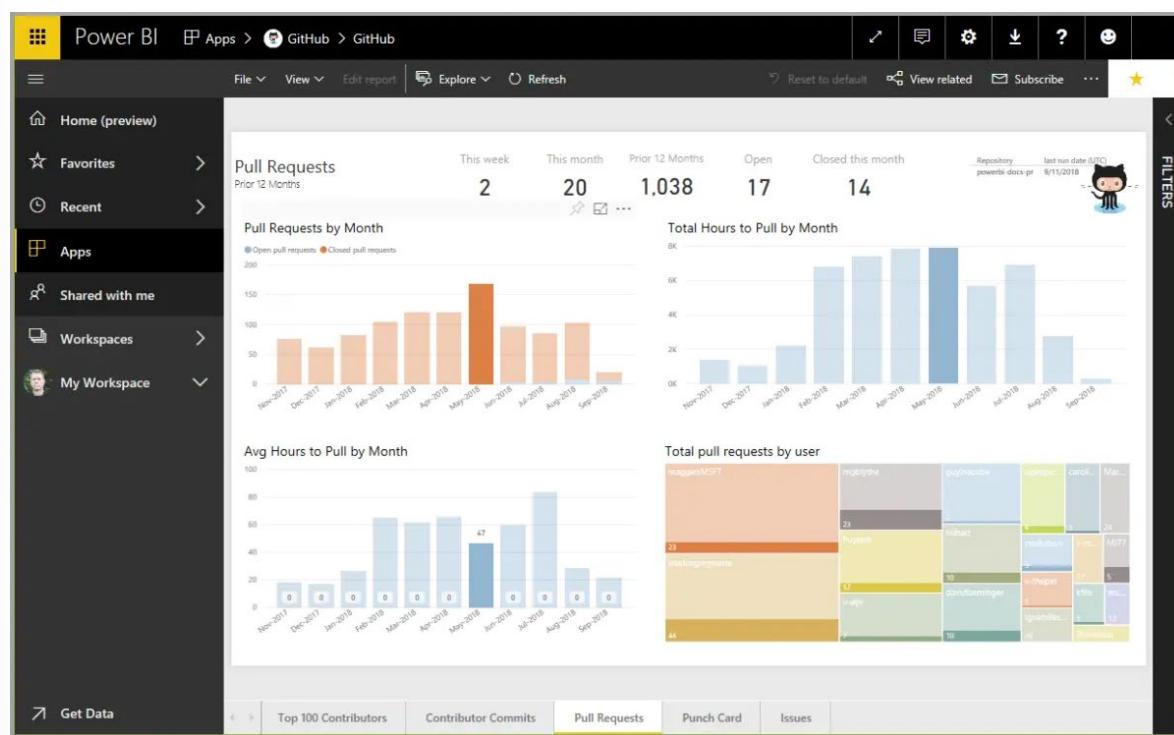
After the data is loaded, the predefined GitHub app dashboard appears.



In addition to the app **dashboard**, the **report** that was generated (as part of the GitHub app) and used to create the dashboard is available, as is the **dataset** (the collection of data pulled from GitHub) that was created during data import and used to create the GitHub report.

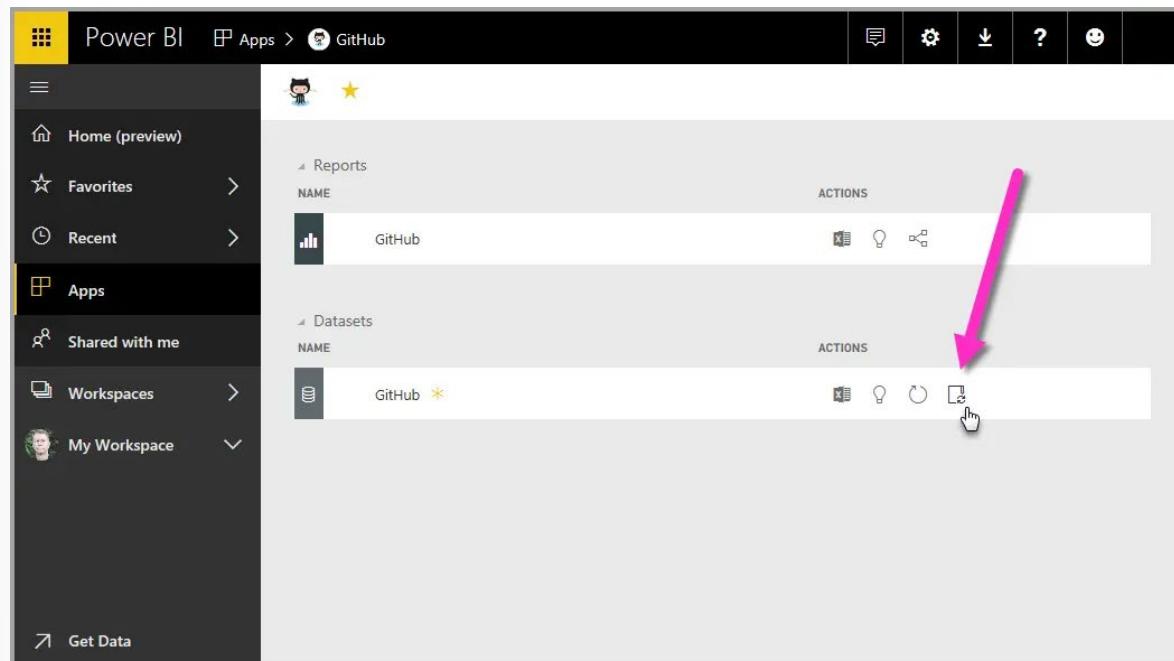


On the dashboard, you can select any of the visuals and interact with them. As you do so, all the other visuals on the page will respond. For example, when the **May 2018** bar is selected in the **Pull Requests (by month)** visual, the other visuals on the page adjust to reflect that selection.

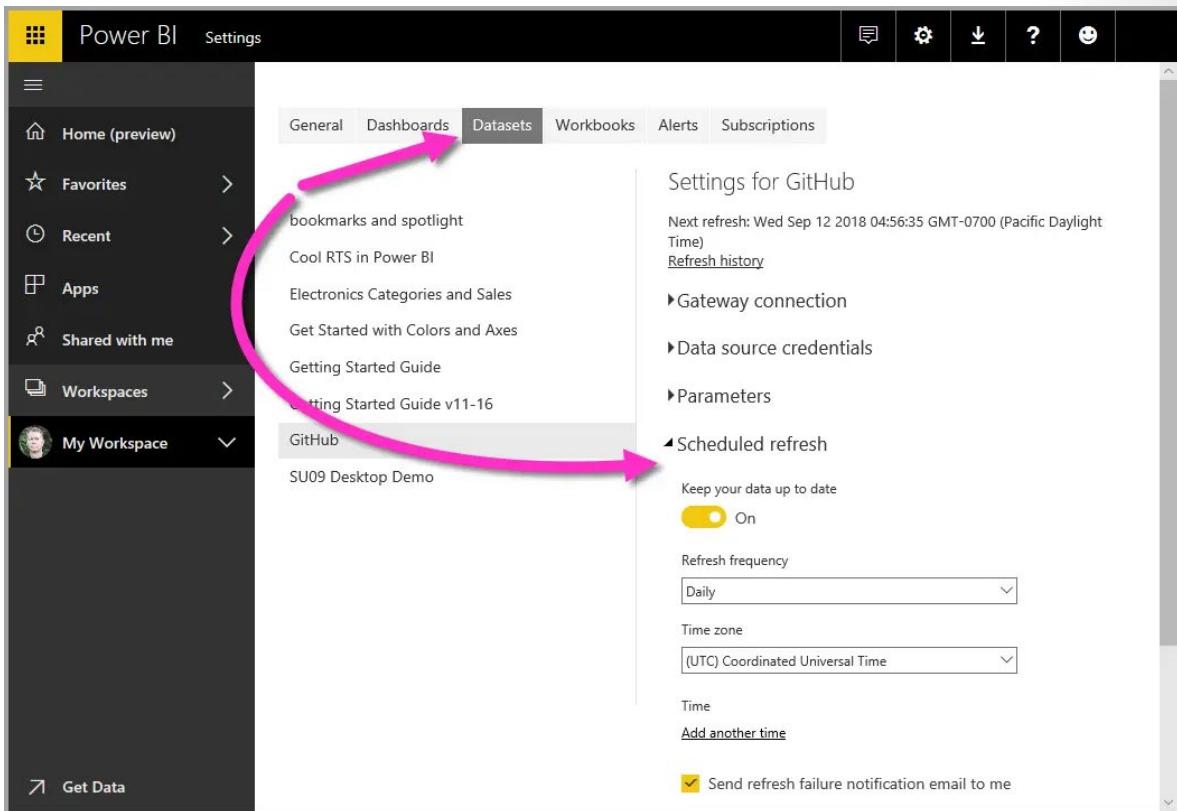


## Update data in the Power BI service

You can also choose to **update** the dataset for an app, or other data that you use in Power BI. To set update settings, select the schedule update icon for the dataset to update, and then use the menu that appears. You can also select the update icon (the circle with an arrow) next to the schedule update icon to update the dataset immediately.



The **Datasets** tab is selected on the **Settings** page that appears. In the right pane, select the arrow next to **Scheduled refresh** to expand that section. The **Settings** dialog box appears on the canvas, letting you set the update settings that meet your needs.



That's enough for our quick look at the Power BI service. There are many more things you can do with the service, and we'll cover these later in this module and in upcoming modules. Also, remember that there are many types of data you can connect to, and all sorts of apps, with more of both coming all the time.

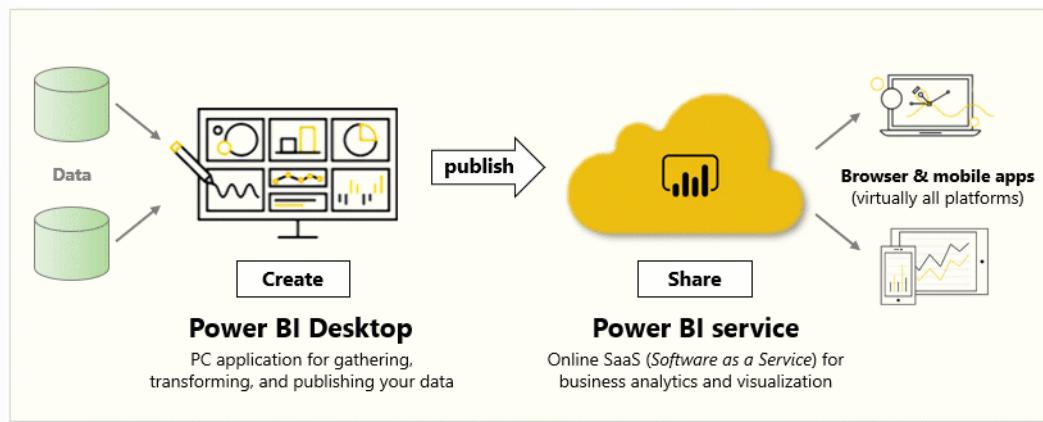
# Get data with Power BI Desktop

## Overview of Power BI Desktop

Power BI Desktop is a free application for PCs that lets you gather, transform, and visualize your data. In this module, you'll learn how to find and collect data from different sources and how to clean or transform it. You'll also learn tricks to make data-gathering easier.

Power BI Desktop and the Power BI Service work together. You can create your reports and dashboards in Power BI Desktop, and then publish them to the Power BI Service for others to consume.

The following are the tasks that you will complete in this module:



**Video:** Introduction to Power BI Desktop



<https://www.microsoft.com/videoplayer/embed/RE3kZ3T>

To perform the exercises in this module, you'll need to have Power BI desktop installed and have a Power BI Service account set up.

## Download Power BI Desktop

You can download Power BI Desktop from the web or as an app from the Microsoft Store on the Windows tab.

| Download Strategy | Link                                                                                                                                 | Notes                             |
|-------------------|--------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------|
| Windows Store App | <b>Windows Store</b> ( <a href="https://aka.ms/pbidesktopstore">https://aka.ms/pbidesktopstore</a> )                                 | Will automatically stay updated   |
| Download from web | <b>Download .msi</b> ( <a href="https://go.microsoft.com/fwlink/?LinkId=521662">https://go.microsoft.com/fwlink/?LinkId=521662</a> ) | Must manually update periodically |

## Sign in to Power BI service

Before you can sign in to Power BI, you'll need an account. To get a free trial, go to [app.powerbi.com<sup>3</sup>](https://app.powerbi.com) and sign up with your email address.

For detailed steps on setting up an account, see [Sign in to Power BI service<sup>4</sup>](#).

## Download sample data

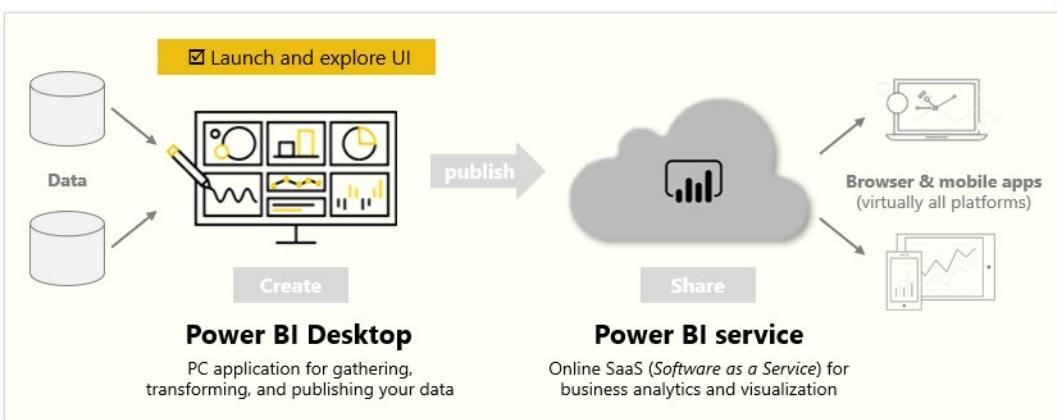
To follow along with the examples in the videos and on the pages, download the sample Excel workbook [Here<sup>5</sup>](#) and import into Power BI Desktop ([Get Data](#) [Excel](#)).

## Explore Power BI Desktop

The idea of building and sharing reports is an abstract concept. It will make more sense if you explore Power BI Desktop hands-on. The first step is to launch and explore the user interface (UI).

In this unit, you will:

- Launch the Power BI Desktop.
- Explore the UI.



**Video:** Overview of Power BI Desktop and the Power BI service



<https://www.microsoft.com/videoplayer/embed/RE3kOEX>

<sup>3</sup> <https://go.microsoft.com/fwlink/?linkid=2101313>

<sup>4</sup> <https://docs.microsoft.com/power-bi/consumer/end-user-sign-in>

<sup>5</sup> <https://go.microsoft.com/fwlink/?linkid=2114225>

**Note:**

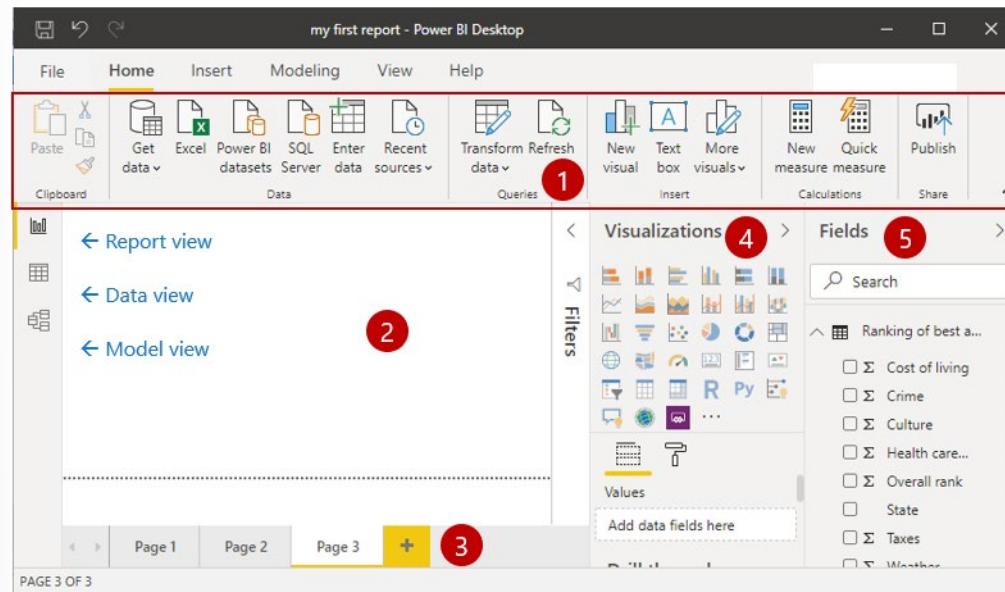
To follow along with the examples in the videos and on this page, download the sample Excel workbook [Here<sup>6</sup>](#) and import into Power BI Desktop (**Get Data** **Excel**) if you haven't already.

## Launch Power BI Desktop

When you launch Power BI Desktop, the **Getting Started** dialog box will appear, which provides useful links to forums, blogs, and introductory videos. Close this dialog box for now, but keep the **Show this screen on startup** option selected so that you can explore it later.

## Explore the report building environment

In Power BI Desktop, you'll begin to build reports in the **Report** view. You'll be working in five main areas:

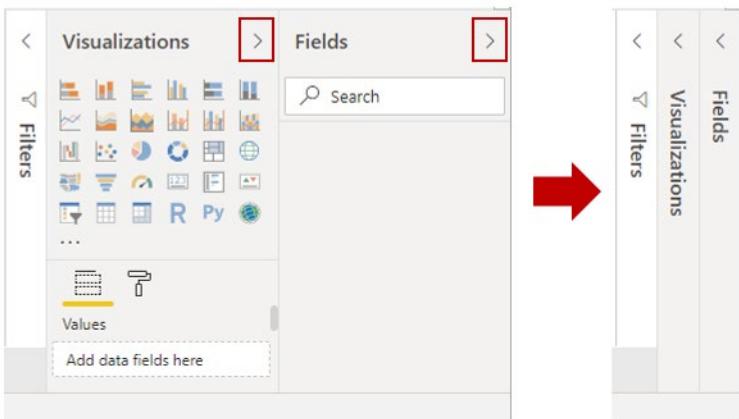


1. **Ribbon** - Displays common tasks that are associated with reports and visualizations.
2. **Report view, or canvas** - Where visualizations are created and arranged. You can switch between **Report**, **Data**, and **Model** views by selecting the icons in the left column.
3. **Pages tab** - Located along the bottom of the page, this area is where you would select or add a report page.
4. **Visualizations pane** - Where you can change visualizations, customize colors or axes, apply filters, drag fields, and more.
5. **Fields pane** - Where query elements and filters can be dragged onto the **Report** view or dragged to the **Filters** area of the Visualizations pane.

**Tip:**

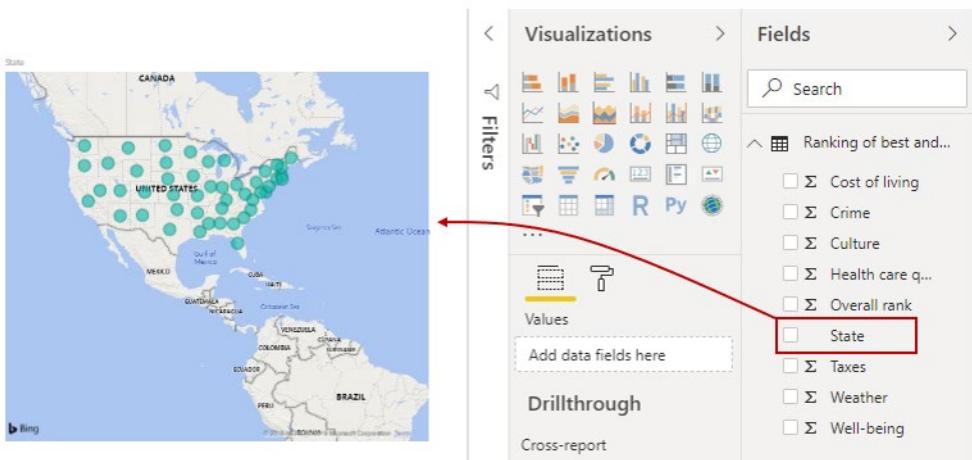
You can collapse the Visualizations and columns panes to provide more space in the **Report** view by selecting the small arrow, as shown in the following screenshot.

<sup>6</sup> <https://go.microsoft.com/fwlink/?linkid=2114225>



## Create a visual

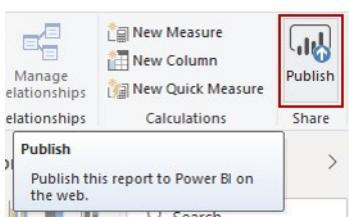
To create a visual, drag a field from the **Fields** list onto the **Report** view.



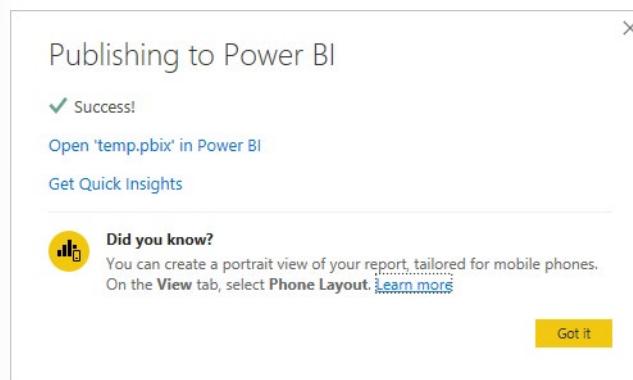
For example, Power BI Desktop automatically created a map-based visualization because it recognized that the **State** field contained geolocation data. (Note that screenshot examples come from a variety of datasets.)

## Publish a report

After creating a report with a few visuals, you're ready to publish to the Power BI service. On the **Home** ribbon on the Power BI Desktop, select **Publish**.



You'll be prompted to sign in to Power BI. When you've signed in and the publish process is complete, the following dialog box will appear. You can select the link below **Success!**, which will take you to the Power BI service, where you can see the report that you published.



## Pin a visual to a dashboard

When you view a published report in the Power BI service, you can choose the **Pin** icon to pin that visual to a dashboard.



You can choose whether to pin the visual to an existing dashboard or to create a new dashboard.

For more information, see [Report View in Power BI Desktop<sup>7</sup>](#).

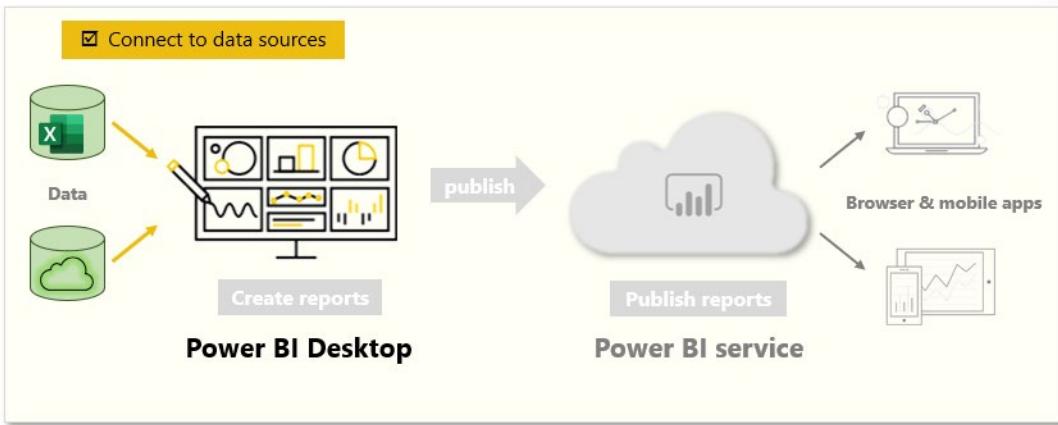
## Connect to data sources

Power BI Desktop connects to many types of data sources, including local databases, worksheets, and data on cloud services. Sometimes when you gather data, it's not quite as structured, or clean, as you want it to be. To structure data, you can transform it, meaning that you can split and rename columns, change data types, and create relationships between columns.

In this unit, you will:

- Connect to data.
- Import data into Power BI Desktop.

<sup>7</sup> <https://docs.microsoft.com/power-bi/desktop-report-view/?azure-portal=true>



**Video:** Connecting to data sources



<https://www.microsoft.com/videoplayer/embed/RE3wRgY>

*Note:*

To follow along with the examples in the videos and on this page, download the sample Access database **Here**<sup>8</sup> and import into Power BI Desktop (**Get Data Database Access database**). If you have any issues loading the Access database, please read this **article**<sup>9</sup>.

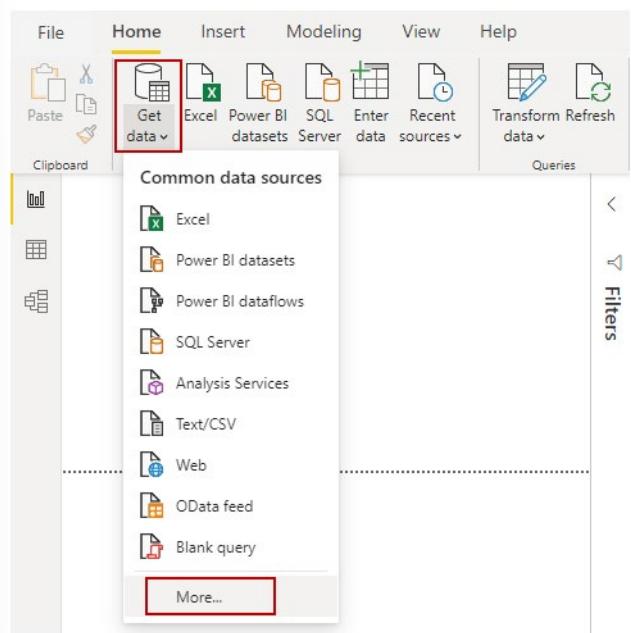
You can connect Power BI Desktop to many types of data sources, including on-premises databases, Microsoft Excel workbooks, and cloud services. Currently, there are about 60 Power BI-specific connectors to cloud services such as GitHub and Marketo. You can also connect to generic sources through XML, CSV, text, and ODBC. Power BI will even extract tabular data directly from a website URL.

## Connect to data

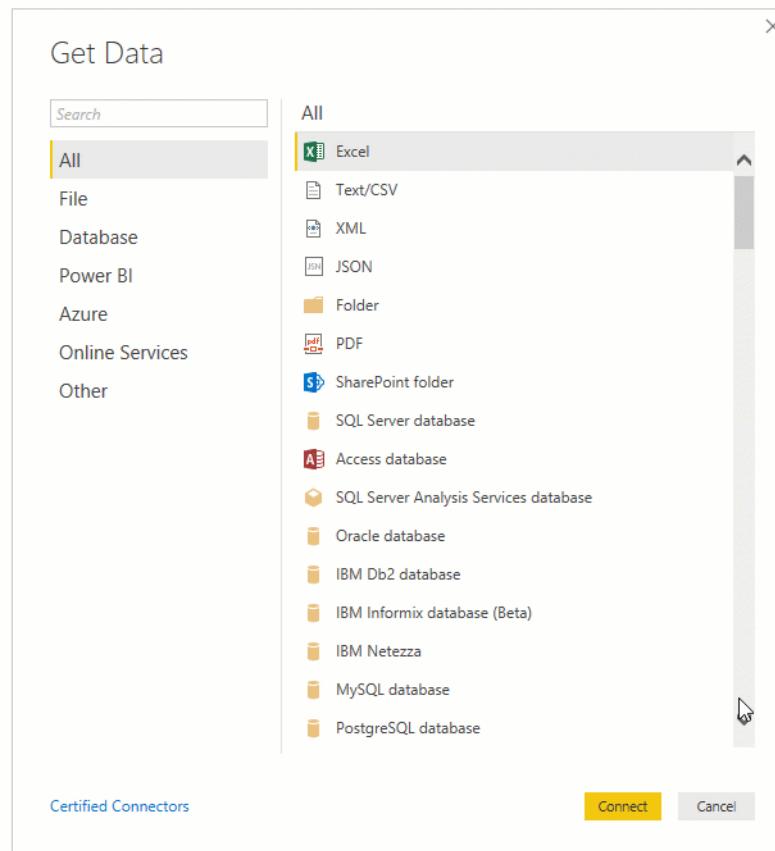
When you start Power BI Desktop, you can choose **Get Data** from the ribbon on the **Home** tab.

<sup>8</sup> <https://go.microsoft.com/fwlink/?linkid=2120368>

<sup>9</sup> <https://go.microsoft.com/fwlink/?linkid=2131277>

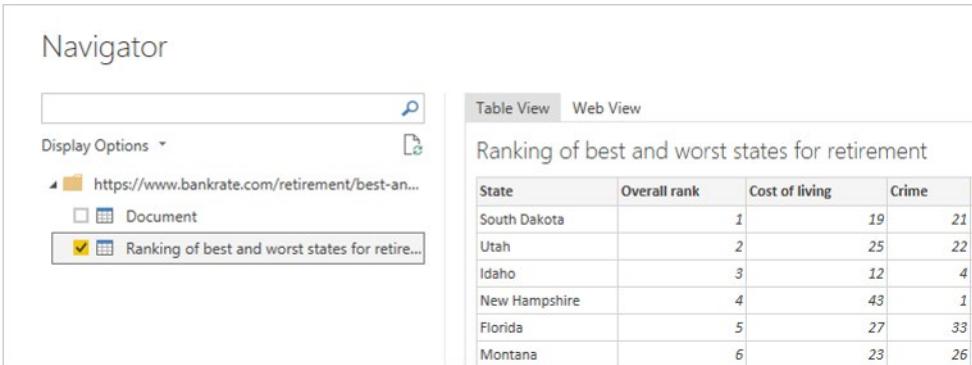


In Power BI Desktop, several types of data sources are available. Select a source to establish a connection. Depending on your selection, you'll be asked to find the source on your computer or network. You might be prompted to sign in to a service to authenticate your request.



## Choose data to import

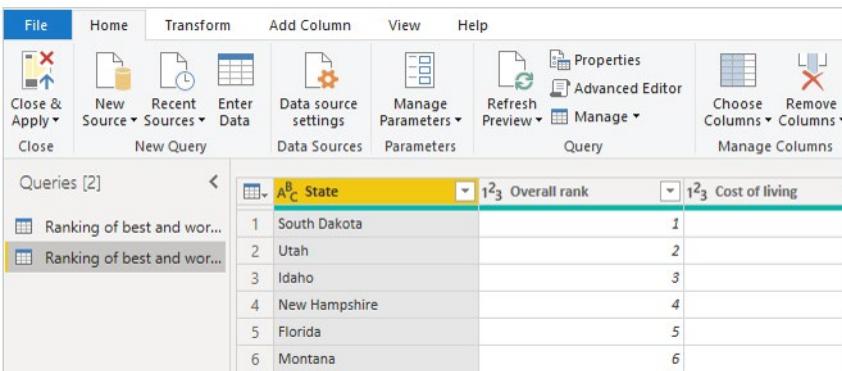
After connecting, the first window that you'll see is the **Navigator**. The **Navigator** window displays the tables or tables of your data source, and selecting a table or table gives you a preview of its contents. You can then import your selected tables or tables immediately by selecting **Load**, or you can select **Transform Data** to transform and clean your data before importing.



The screenshot shows the Power BI Navigator window. On the left, there's a search bar and a 'Display Options' dropdown. Below that is a tree view showing a folder containing a document and a selected table named 'Ranking of best and worst states for retire...'. On the right, under 'Table View', is a preview of a table titled 'Ranking of best and worst states for retirement'. The table has columns: State, Overall rank, Cost of living, and Crime. The data rows are: South Dakota (rank 1), Utah (rank 2), Idaho (rank 3), New Hampshire (rank 4), Florida (rank 5), and Montana (rank 6).

| State         | Overall rank | Cost of living | Crime |
|---------------|--------------|----------------|-------|
| South Dakota  | 1            | 19             | 21    |
| Utah          | 2            | 25             | 22    |
| Idaho         | 3            | 12             | 4     |
| New Hampshire | 4            | 43             | 1     |
| Florida       | 5            | 27             | 33    |
| Montana       | 6            | 23             | 26    |

After you've selected the tables that you'd like to bring into Power BI Desktop, select the **Load** button. You might want to make changes to those tables before you load them. For example, if you only want a subset of customers or a specific country or region, select the **Edit** button and filter data before loading.



The screenshot shows the Power BI Home tab. The ribbon includes File, Home, Transform, Add Column, View, and Help. Under the Home tab, there are buttons for Close & Apply, New Source, Recent Sources, Enter Data, Data source settings, Manage Parameters, Refresh Preview, Properties, Advanced Editor, Choose Columns, Remove Columns, and Manage Columns. Below the ribbon, the 'Queries [2]' section lists two queries: 'Ranking of best and wor...' and 'Ranking of best and wor...'. The main area shows a table with columns: State, Overall rank, and Cost of living. The data rows are identical to the ones in the Navigator preview: South Dakota (rank 1), Utah (rank 2), Idaho (rank 3), New Hampshire (rank 4), Florida (rank 5), and Montana (rank 6).

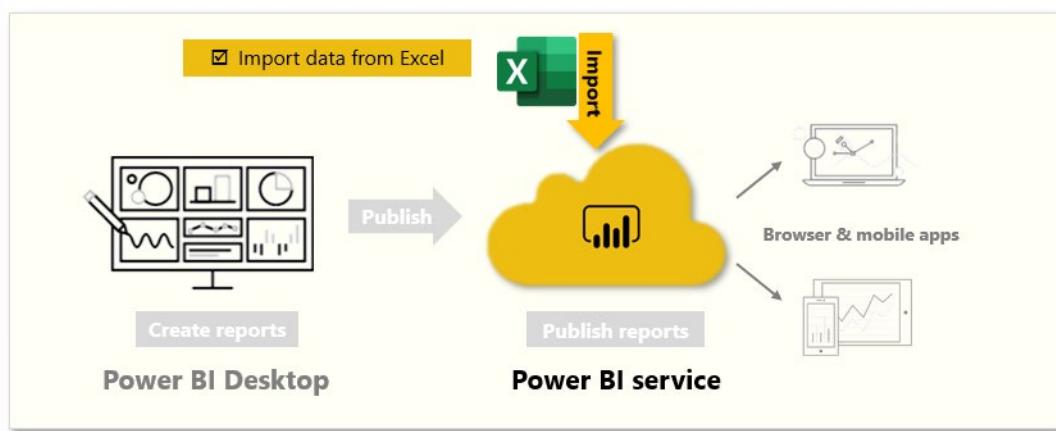
| State         | Overall rank | Cost of living |
|---------------|--------------|----------------|
| South Dakota  | 1            |                |
| Utah          | 2            |                |
| Idaho         | 3            |                |
| New Hampshire | 4            |                |
| Florida       | 5            |                |
| Montana       | 6            |                |

No matter what type of data you need, you're likely to find a way to import it into Power BI Desktop.

## Get data from Excel

Likely, you've used Microsoft Excel to create or view reports or to build pie charts or other visuals. Getting your Excel data into Power BI is a straightforward process.

In this unit, you will bring Excel workbooks into Power BI.

**Video:**

<https://www.microsoft.com/videoplayer/embed/RE3nkKG>

**Note:**

To follow along with the example in this video, download the sample Excel workbook [Here<sup>10</sup>](#).

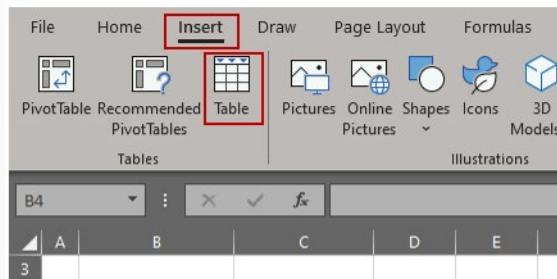
This unit explains how you can import an Excel workbook file that contains a simple table from a local drive into Power BI. You'll then learn how to begin exploring that table's data in Power BI by creating a report.

**Note:**

Up until now, we've been importing data through Power BI Desktop. This unit page is done from the Power BI service.

## Make sure that your data is formatted as a table

For Power BI to import data from your workbook, that data needs to be formatted as a table. In Excel, you can highlight a range of cells, and then on the **Insert** tab of the Excel ribbon, select **table**.

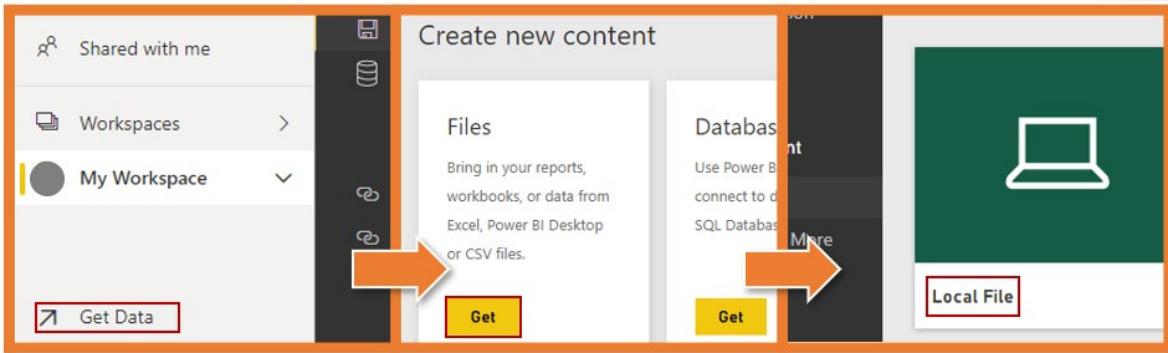


<sup>10</sup> <https://go.microsoft.com/fwlink/?linkid=2114225>

Make sure that each column has a good name; it will make it easier for you to find the data that you want when creating your reports in Power BI.

## Import from a local drive

Wherever you keep your files, Power BI makes importing them simple. In Power BI, you can go to **Get Data** **Files** **Local File** to select the Excel file that you want.

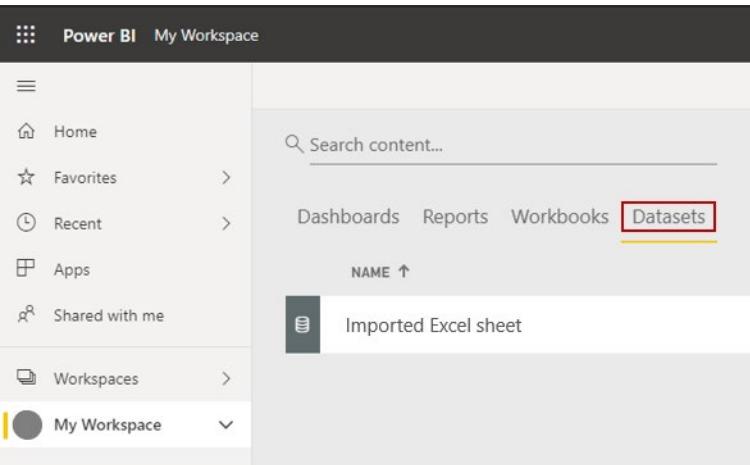


After the file has been imported into Power BI, you can begin creating reports.

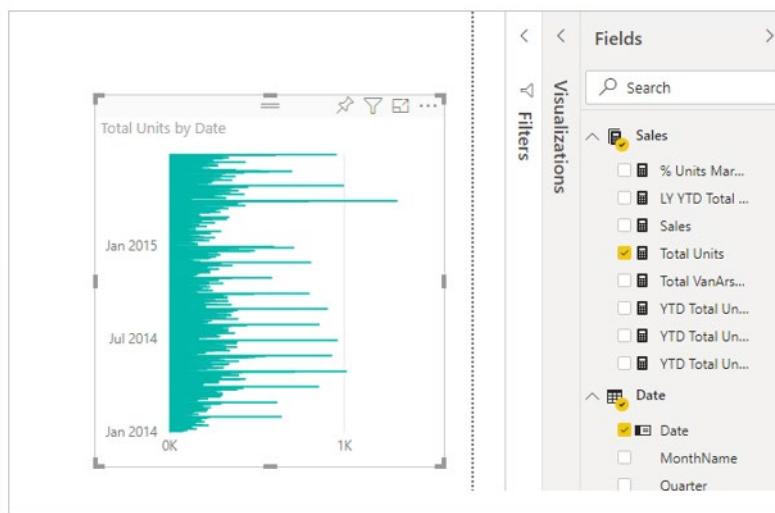
Your files don't have to be on a local drive. If you save your files on OneDrive or SharePoint Team Site, that's even better.

## Create reports

After your workbook's data has been imported, a dataset is created in Power BI and it will appear under **Datasets**.



Now, you can begin exploring your data by creating reports and dashboards. Select the **Open menu** icon next to the dataset and then select **Explore**. A new blank report canvas appears. On the right-hand side, under **Fields**, are your tables and columns. Select the fields for which you want to create a new visualization on the canvas.



You can change the type of visualization and apply filters and other properties under **Visualizations**.

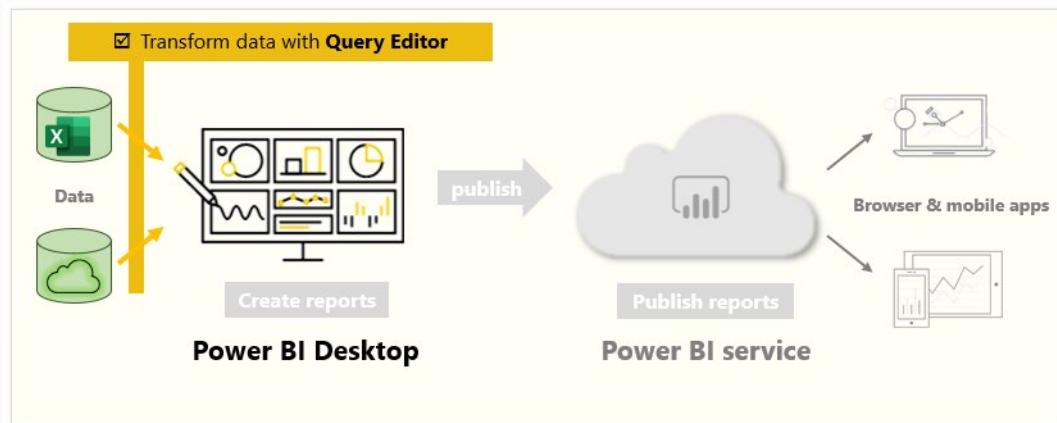
If you use any of Excel's advanced BI features like **Power Query**, **Power Pivot**, or **Power View**, you can import that data into Power BI, too.

For more information, see [Get data from Excel workbook files<sup>11</sup>](#).

## Transform data to include in a report

Sometimes, your data might contain extra data or have data in the wrong format. Power BI Desktop includes the **Power Query Editor** tool, which can help you shape and transform data so that it's ready for your models and visualizations.

In this unit, you will transform data with Power Query Editor.



**Video:** Transform data

<sup>11</sup> <https://docs.microsoft.com/power-bi/service-excel-workbook-files>



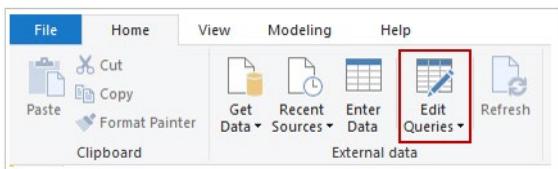
<https://www.microsoft.com/en-us/videoplayer/embed/RE3x9uw>

*Note:*

To follow along with the examples in the videos and on this page, download the sample Access database [Here<sup>12</sup>](#) and import into Power BI Desktop (**Get Data Database Access database**). If you have any issues loading the Access database, please read this [article<sup>13</sup>](#).

## Launch Power Query Editor

To begin, select **Transform** from the **Navigator** window to launch Power Query Editor. You can also launch Power Query Editor directly from Power BI Desktop by using the **Transform data** button on the Home ribbon.



After loading your data into Power Query Editor, you'll see the following screen.

| A-C State      | Overall rank |
|----------------|--------------|
| South Dakota   | 1            |
| Utah           | 2            |
| Idaho          | 3            |
| New Hampshire  | 4            |
| Florida        | 5            |
| Montana        | 6            |
| North Carolina | 6            |
| Wyoming        | 8            |
| Nebraska       | 9            |

1. In the ribbon, the active buttons enable you to interact with the data in the query.
2. On the left pane, queries (one for each table, or entity) are listed and available for selecting, viewing, and shaping.
3. On the center pane, data from the selected query is displayed and available for shaping.

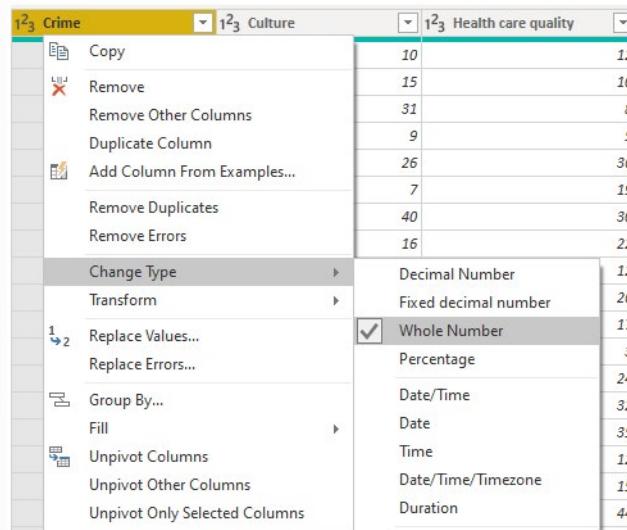
<sup>12</sup> <https://go.microsoft.com/fwlink/?linkid=2120368>

<sup>13</sup> <https://go.microsoft.com/fwlink/?linkid=2131277>

4. The **Query Settings** window lists the query's properties and applied steps.

## Transform data

On the center pane, right-clicking a column displays the available transformations. Examples of the available transformations include removing a column from the table, duplicating the column under a new name, or replacing values. From this menu, you can also split text columns into multiples by common delimiters.

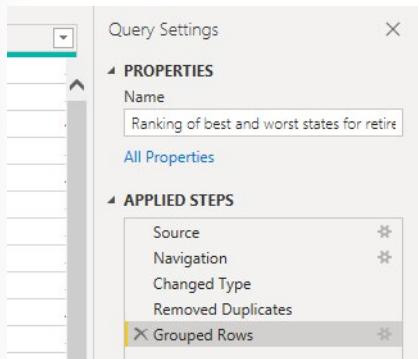


The Power Query Editor ribbon contains additional tools that can help you change the data type of columns, add scientific notation, or extract elements from dates, such as day of the week.

*Tip:*

If you make a mistake, you can undo any step from the **Applied Steps** list.

As you apply transformations, each step appears in the **Applied Steps** list on the Query Settings pane. You can use this list to undo or review specific changes, or even change the name of a step. To save your transformations, select **Close & Apply** on the **Home** tab.



After you select **Close & Apply**, Power Query Editor applies the query changes and applies them to Power BI Desktop.

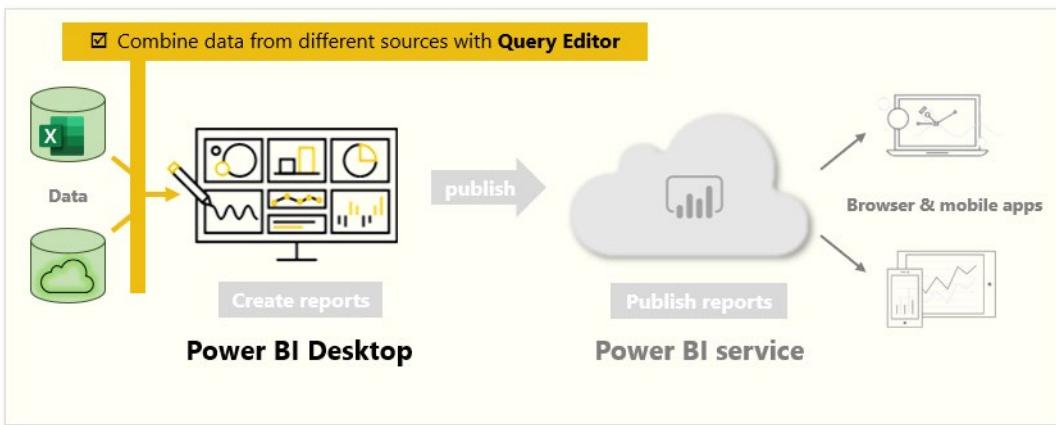
For more information, see **Quickstart: Using Power Query in Power BI Desktop**<sup>14</sup>.

<sup>14</sup> <https://docs.microsoft.com/power-query/power-query-quickstart-using-power-bi/?azure-portal=true>

## Combine data from multiple sources

With Power BI Desktop, you can use the Power Query Editor tool to combine data from multiple sources into a single report.

In this unit, you will combine data from different sources by using Query Editor.



**Video:** Modeling and transforming data



<https://www.microsoft.com/en-us/videoplayer/embed/RE3wTTj>

**Note:**

To follow along with the examples in the videos and on this page, download the sample Access database [here<sup>15</sup>](#) and import into Power BI Desktop (**Get Data Database Access database**). The **FactData1** folder is [here<sup>16</sup>](#). If you have any issues loading the Access database, please read this [article<sup>17</sup>](#).

## Add more data sources

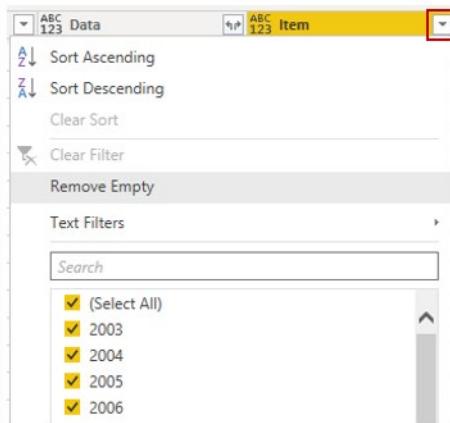
To add more sources to an existing report, from the Home ribbon, select **Transform data** and then select **New Source**. You can use many potential data sources in Power BI Desktop, including folders. By connecting to a folder, you can import data from multiple Excel or CSV files at once.

Power Query Editor allows you to apply filters to your data. For example, selecting the drop-down arrow next to a column opens a checklist of text filters. Clearing a filter allows you to remove values from your model before the data is loaded into Power BI.

<sup>15</sup> <https://go.microsoft.com/fwlink/?linkid=2120368>

<sup>16</sup> <https://go.microsoft.com/fwlink/?linkid=2124018>

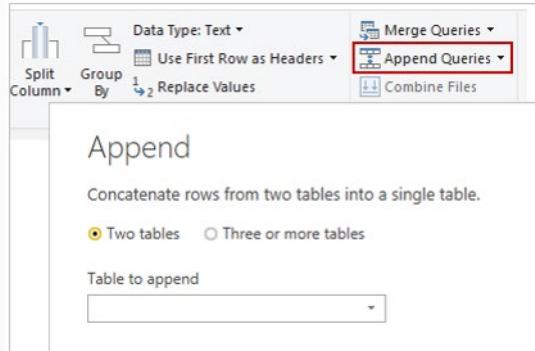
<sup>17</sup> <https://go.microsoft.com/fwlink/?linkid=2131277>



**Important:** Filtering in the Power Query Editor changes which data is loaded into Power BI. Later, when you apply filters in the Data View or Report View, those filters only apply to what you see in visuals but do not change the underlying dataset.

## Merge and append queries

You can also merge and append queries. In other words, Power BI pulls data that you select from multiple tables or various files into a single table. Use the Append Queries tool to add the data from a new table to an existing query. Power BI Desktop attempts to match the columns in your queries, which you can then adjust as necessary in Power Query Editor.



## Write customized queries

You can use the Add Custom Column tool to write new customized query expressions by using the powerful M language.



For more information, see [Tutorial: Shape and combine data in Power BI Desktop<sup>18</sup>](#).

## Clean data to include in a report

While Power BI can import your data from almost any source, its visualization and modeling tools work best with columnar data. Sometimes, your data won't be formatted in simple columns, which is often the case with Excel spreadsheets.

In this unit, you will clean columnar data with Power Query Editor.

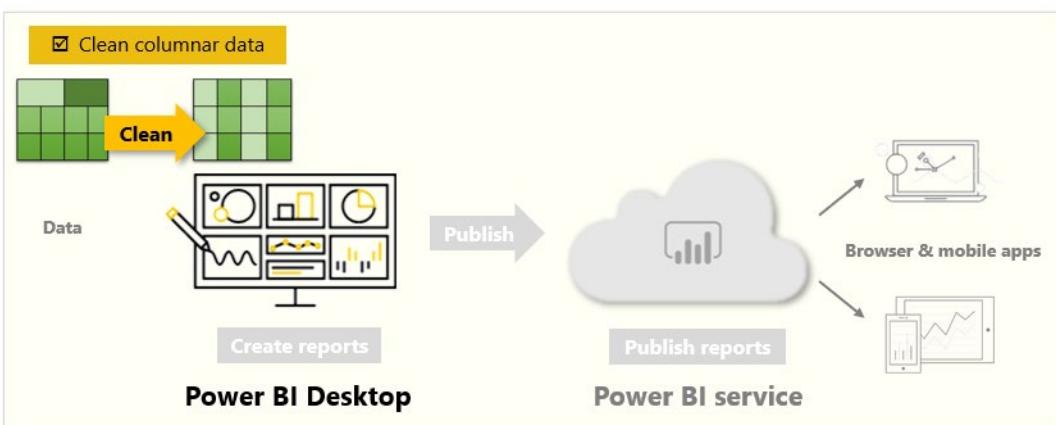
**Video:**



<https://www.microsoft.com/en-us/videoplayer/embed/RE3wRhf>

**Note:**

To follow along with the Excel spreadsheet in this video, download the .xlsx [Here<sup>19</sup>](#)



<sup>18</sup> <https://docs.microsoft.com/power-bi/desktop-shape-and-combine-data/?azure-portal=true>

<sup>19</sup> <https://go.microsoft.com/fwlink/?linkid=2123759>

A table layout that looks good to the human eye might not be optimal for automated queries. For example, the following spreadsheet has headers that span multiple columns.

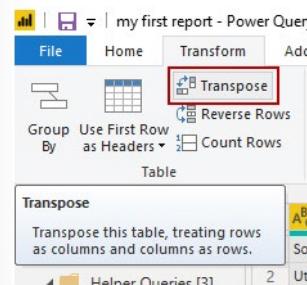
| 2R x 9C |      | Seattle |             | Portland      |       |             |  |
|---------|------|---------|-------------|---------------|-------|-------------|--|
| 1       | 2    | Bikes   | Accessories | Miscellaneous | Bikes | Accessories |  |
| 3       | 2005 | 33323   | 13394       | 4455          | 33323 | 13394       |  |
| 4       | 2006 | 55342   | 19983       | 5563          | 55342 | 19983       |  |
| 5       | 2007 | 33234   | 18884       | 3348          | 33234 | 18884       |  |
| 6       | 2008 | 33252   | 19893       | 2239          | 33252 | 19893       |  |
| 7       | 2009 | 22332   | 18840       | 2232          | 22332 | 18840       |  |
| 8       | 2010 | 23331   | 18890       | 4343          | 23331 | 18890       |  |
| 9       | 2011 | 33532   | 18790       | 3434          | 33532 | 18790       |  |
| 10      | 2012 | 11001   | 11000       | 8840          | 11001 | 11000       |  |
| 11      | 2013 | 10221   | 9900        | 8892          | 10221 | 9900        |  |

## Clean data

Fortunately, Power Query Editor has tools to help you quickly transform multi-column tables into datasets that you can use.

## Transpose data

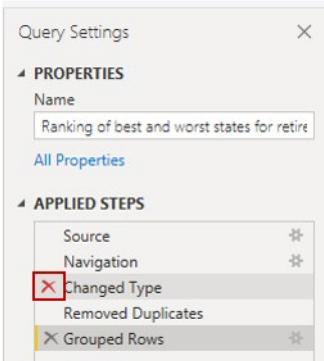
By using **Transpose** in Power Query Editor, you can swap rows into columns to better format the data.



## Format data

You might need to format data so that Power BI can properly categorize and identify that data. With some transformations, you'll cleanse data into a dataset that you can use in Power BI. Examples of powerful transformations include promoting rows into headers, using **Fill** to replace *null* values, and **Unpivot columns**.

With Power BI, you can experiment with transformations and determine which will transform your data into the most usable columnar format. Remember, the **Applied Steps** section of Power Query Editor records all your actions. If a transformation doesn't work the way that you intended, select the **X** next to the step, and then undo it.



After you've cleaned your data into a usable format, you can begin to create powerful visuals in Power BI.

For more information, see **Tutorial: Combine sales data from Excel and an OData feed<sup>20</sup>**.

## Summary

Congratulations, you now know how to get data into Power BI Desktop and how to shape or transform that data so you can create compelling visuals.

<sup>20</sup> <https://docs.microsoft.com/power-bi/desktop-tutorial-analyzing-sales-data-from-excel-and-an-odata-feed/?azure-portal=true>

# Create reports with Power BI for Teams

## Introduction

Data empowers employees to make better decisions, accelerate business transformation, and grow the organization. Providing business intelligence reports and dashboards offers users a way to visualize, interact, and share insights with others. Microsoft Power BI and Dataverse for Teams embeds powerful self-service reports and dashboards so that users can access your organization's data and insights without having to leave Microsoft Teams.

The learning objectives for this module are:

- Import a Microsoft Dataverse dataset into Power BI.
- Create a new Power BI report based on the dataset.
- Edit a report.
- Share a report with others.

[!NOTE]

You must have the appropriate Power BI license to build and share Power BI reports. For more information about Power BI licenses, see the links that are provided in the Summary unit at the end of this module.

[!Note]

For this learning module you will need to have provisioned a Dataverse for Teams environment. Please see the links that are provided in the Summary unit at the end of the module for resources on how to provision the environment.

## Dataverse for Teams sample data

In order to show how to import data from Dataverse for Teams into Power BI, we need to first have some data in Dataverse for Teams. For the purposes of this learning path, we will use a table called **Cost Trends** that stores information captured by a Power App. You can use the **Cost trends<sup>21</sup>** spreadsheet to upload data. Select **Download** and then extract the spreadsheet to your local computer. Select a Microsoft team that has a Dataverse for Teams environment then create a new table named **Cost Trends** with the fields and data types that are shown in the following table.

Note: Please see the links that are provided in the Summary unit at the end of the module on how to create tables in Dataverse for Teams.

| Column name      | Data type |
|------------------|-----------|
| Name             | Text      |
| Assembly         | Number    |
| Date             | Date      |
| Materials        | Number    |
| Package Total    | Number    |
| Printing         | Number    |
| Production Total | Number    |
| Shipping         | Number    |

<sup>21</sup> <https://github.com/MicrosoftDocs/mslearn-developer-tools-power-platform/blob/master/power-apps/dataverse-power-bi/cost-trends.zip>

After creating the column names and data types in **Columns**, there are two ways to enter the sample data into the table. One way is to enter it directly in the table, or you can open Excel and edit the sample data.

To enter the sample data into the table, select **Tables** and then select the ellipsis(...) for the table. Select **Edit data** for the visual table editor experience.

The screenshot shows the Microsoft Power BI service interface. On the left, there's a navigation pane with 'All', 'Apps', 'Chatbots', 'Cloud flows', and a red box highlighting 'Tables'. Under 'Tables', there's a list with 'Cost Trends Sample' expanded, showing 'Columns', 'Relationships', and 'Views'. To the right, the main area shows a table named 'Cost Trends Sample' with columns: Name, Assembly, Date, Materials, Package..., Printing, Product..., and Shipping. The table has 8 rows of data from Jan 2016 to Jul 2016. At the top of the main area, there are buttons for 'New', 'Open in Power Apps', 'Edit', 'Delete table', 'Edit data', and 'Manage permissions'. Below the table, there's a context menu with options: 'Edit', 'Delete table', 'Edit data in Excel' (which is highlighted with a red box), and 'Manage permissions'.

You will need to manually input the data. Copy and paste is not currently an option. Save the table.

The screenshot shows a Microsoft Teams window titled 'Cost Trends'. It displays a table with columns: Name, Assembly, Date, Materials, Package..., Printing, Product..., and Shipping. The table has 8 rows of data from Jan 2016 to Jul 2016. At the top of the table, there are buttons for '+ Add row', '+ Add column', 'Show/hide columns', and 'Refresh'. On the far right, there are buttons for 'Saved' and 'Default'. A vertical sidebar on the left shows 'Activity', 'Chat', 'Teams', 'Calendar', 'Calls', 'Files', 'Power Apps', and '...'. A red box highlights the 'Edit data in Excel' button in the context menu on the right side of the table.

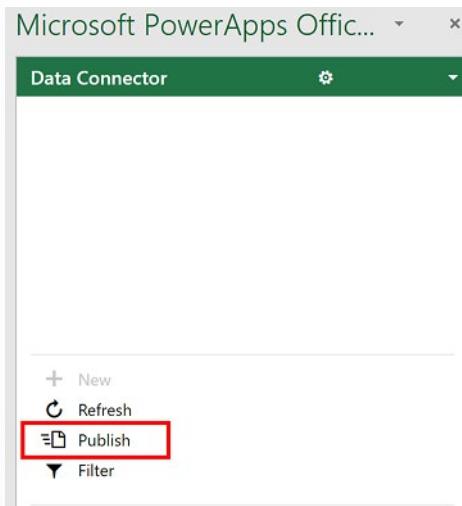
To enter the sample data in Excel, follow the next few steps.

1. Select **Tables**, select the ellipses (...) and select **Edit data in Excel**.

The screenshot shows the Microsoft Power BI service interface again. The 'Tables' section is highlighted with a red box. The 'Cost Trends Sample' table is selected. A context menu on the right shows 'Edit', 'Delete table', 'Edit data in Excel' (highlighted with a red box), 'Edit data', and 'Manage permissions'.

2. A successful Excel notification will appear at the top of the screen. Open the downloaded file in **Excel**.

3. There will be additional columns listed in the spreadsheet. Do not delete those columns. Reorder the columns to match the first eight columns on the CostTrends spreadsheet.
4. Enter the sample data into the table, and save the table.
5. To push the data back to Dataverse, navigate to the **Data Connector** section in the lower right, and select **Publish**.



## Connect to and transform Dataverse for Teams data

To connect to Dataverse for Teams data, you need to install Microsoft Power BI Desktop and then get the environment URL.

### Install Power BI Desktop

Power BI Desktop is a free application that allows you to connect, transform, and visualize your data. It can connect to many data sources, including Microsoft SQL, Excel, SharePoint lists, and Dataverse for Teams tables. Then, Power BI Desktop will transform and clean the data to create interactive reports.

Two ways to install Power BI Desktop are:

- **Windows 10** - Use the **Microsoft Store**<sup>22</sup>. Software updates will automatically install each month.
- **Windows 8 and 10** - Download the installer from <https://powerbi.microsoft.com/desktop><sup>23</sup>. Each month, you'll be prompted to install the latest updates to the application.

### Get the environment URL for the Dataverse for Teams table

The environment URL for the Dataverse for Teams table is required when you are connecting with Power BI Desktop. To find the environment URL, follow these steps:

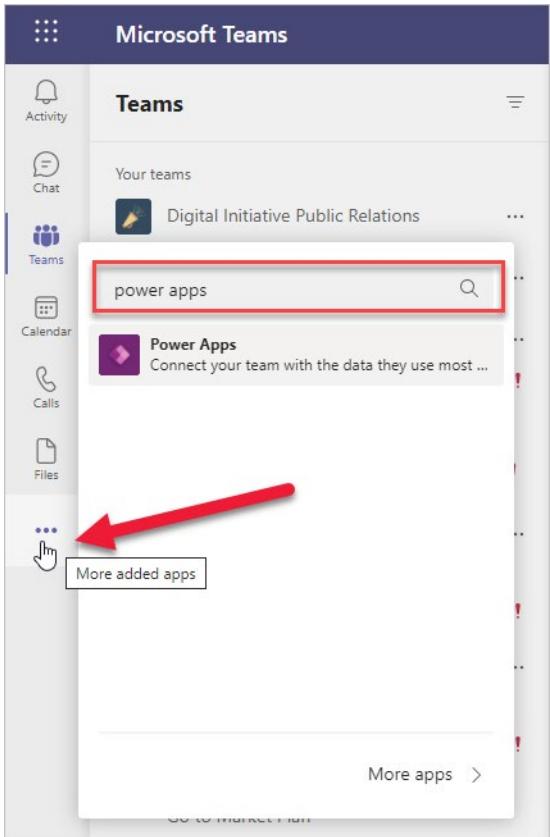
1. Launch Teams or use your browser to open <https://teams.microsoft.com><sup>24</sup>.

<sup>22</sup> <https://www.microsoft.com/store/productId/9NTXR16HNW1T/?azure-portal=true>

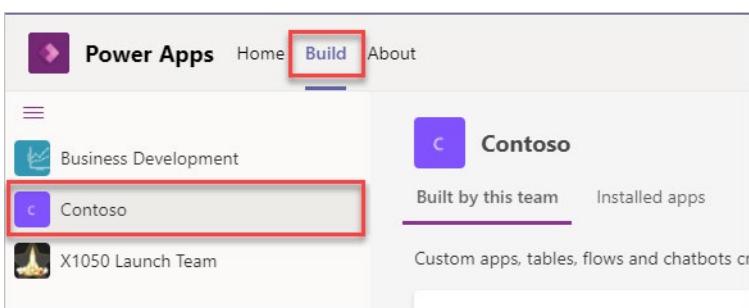
<sup>23</sup> <https://powerbi.microsoft.com/desktop/?azure-portal=true>

<sup>24</sup> <https://teams.microsoft.com/?azure-portal=true>

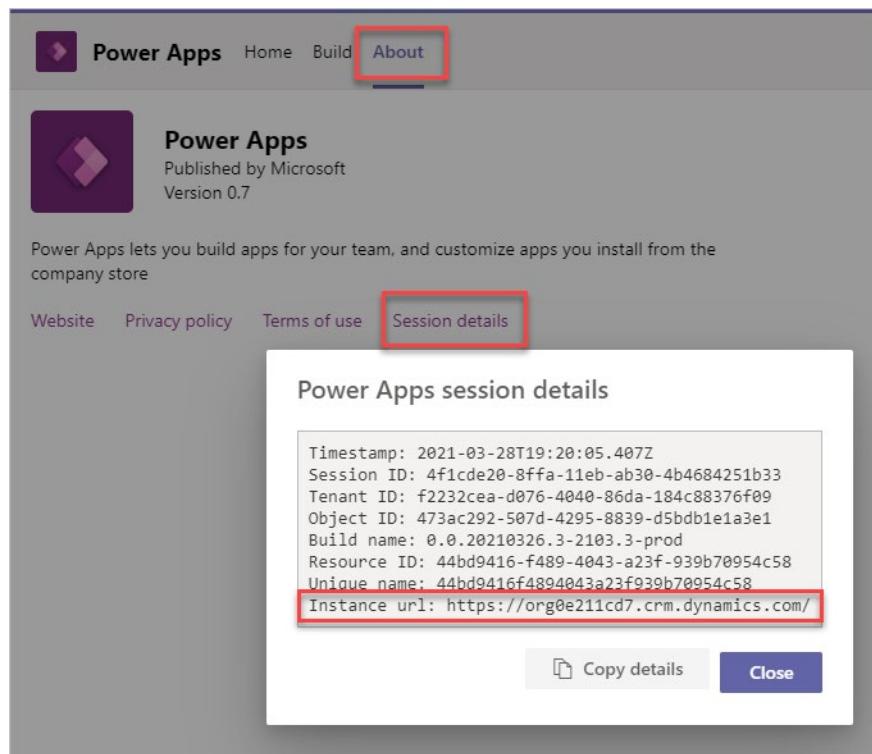
2. Select the **More added apps** ellipsis then type in **power apps** in the search bar.
3. Select **Power Apps** from the search results. If you don't already have it installed click, on **More apps >** and install it from the Teams app store.



4. Select the **Build** tab.
5. Select the team that contains the Dataverse for Teams table.



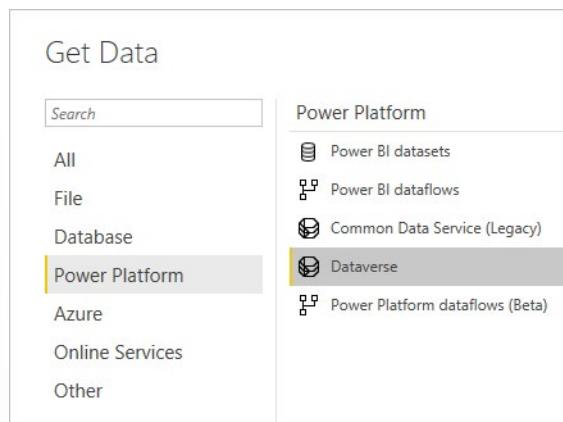
6. Select the **About** tab and then select **Session details**. Copy the **Instance url** into notepad for later reference. The **Instance url** is the environment URL to connect Power BI Desktop to the Dataverse for Teams table.



## Connect to Dataverse for Teams table data

To connect to Dataverse for Teams table data, follow these steps:

1. Launch Power BI Desktop.
2. From the **Home** ribbon, select **Get Data**.
3. Select **Microsoft Power Platform** then **Dataverse**. Select **Connect** to continue.



4. Enter the environment URL for the Dataverse for Teams table that you copied previously. Remove the **https://** protocol and the trailing slash (/) from the URL. Set **Data Connectivity mode** to **Import** and then select **OK**.



5. Select **Sign in** and then use your credentials when prompted. After you've successfully signed in, select **Connect**.
6. In the **Navigator** screen, enter the name of the table to locate it. Select the table by selecting the check box next to the name. For this example, select the Cost Trends table that you created previously.

The screenshot shows the 'Navigator' screen. A search bar at the top contains 'cost'. Below it, a tree view shows 'org0e211cd7.crm.dynamics.com' expanded, with 'cr71f\_costtrends' selected and checked. To the right is a table titled 'cr71f\_costtrends' with columns 'cr71f\_costtrendsid', 'createdon', and 'createdby'. The table lists several rows of data. At the bottom right of the screen are 'Load', 'Transform Data', and 'Cancel' buttons, with 'Transform Data' highlighted by a red box.

| cr71f_costtrendsid                   | createdon             | createdby             |
|--------------------------------------|-----------------------|-----------------------|
| 8EA1391A-858E-EB11-A812-000D3A9A4241 | 3/26/2021 10:46:29 PM | 3BF407BC-848E-EB11-A8 |
| 1CFB077A-858E-EB11-A812-000D3A9A4241 | 3/26/2021 10:49:10 PM | 3BF407BC-848E-EB11-A8 |
| 64B14996-858E-EB11-A812-000D3A9A4241 | 3/26/2021 10:49:57 PM | 3BF407BC-848E-EB11-A8 |
| 95C71724-8A8E-EB11-A812-000D3A9A4241 | 3/26/2021 11:22:33 PM | 3BF407BC-848E-EB11-A8 |
| C88CF53C-8A8E-EB11-A812-000D3A9A4241 | 3/26/2021 11:23:15 PM | 3BF407BC-848E-EB11-A8 |
| D3AEF74F-8A8E-EB11-A812-000D3A9A4241 | 3/26/2021 11:23:51 PM | 3BF407BC-848E-EB11-A8 |
| C882155C-8A8E-EB11-A812-000D3A9A4241 | 3/26/2021 11:24:16 PM | 3BF407BC-848E-EB11-A8 |

7. Select **Transform Data** to open the table in Power Query Editor.

Power Query Editor allows you to clean the data and helps make it easier for you to create charts, graphs, and other visualizations. The following steps use Power Query Editor to transform the data.

## Transform the data

Transforming data is the process of changing data to make it easier to work with. For example, you might want to remove unnecessary columns, change data types, or filter the data before creating Power BI reports. Dataverse for Teams tables have system columns that are used by Dataverse for Teams. You don't necessarily need these system columns when you are creating Power BI visualizations.

For the Cost Trends table data, you need to remove unnecessary columns, rename columns, and verify that the correct data types are used:

1. To reduce the number of columns in the Power Query Editor **Home** ribbon, select **Choose Columns**.

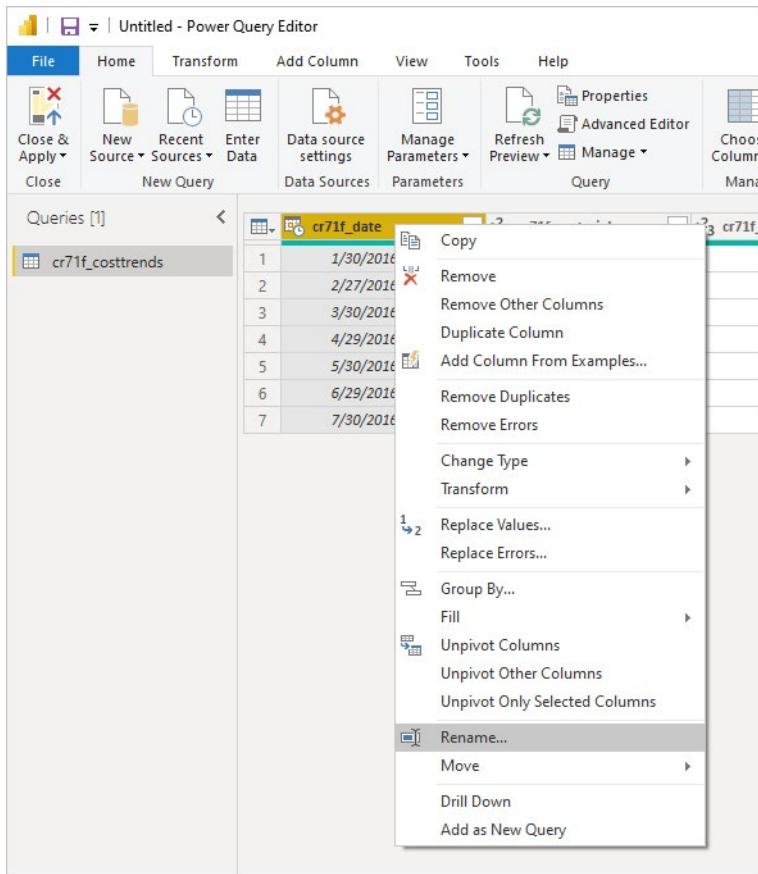
The screenshot shows the Power Query Editor interface. The ribbon has 'Home' selected. In the 'Transform' group, the 'Choose Columns' button is highlighted with a red box. A 'Choose Columns' dialog box is open on the right, titled 'Choose Columns'. It says 'Choose the columns to keep' and lists four items: '(Select All Columns)', 'cr71f\_costtrendid', 'createdon', and 'createdby', all with checked checkboxes. The main area shows a table with columns 'A<sup>B</sup> cr71f\_costtrendid', 'B<sup>C</sup> createdon', and 'A<sup>B</sup> createdby'. The table has 7 rows of data. The 'Query Settings' pane on the right shows 'PROPERTY Name' set to 'cr71f\_costtrends' and 'APPLIED Source' set to 'Navigation'.

2. Select only the following columns. The prefix is different in your environment.

- cr71f\_date
- cr71f\_materials
- cr71f\_printing
- cr71f\_assembly
- cr71f\_productiontotal
- cr71f\_shipping
- cr71f\_packagetotal

Select **OK** to continue.

3. Rename columns by right-clicking the column name and selecting **Rename**.



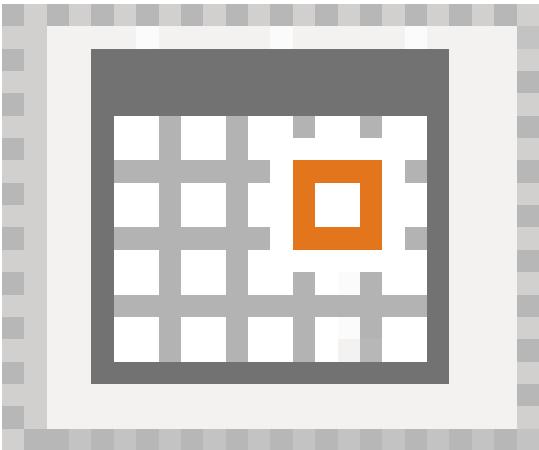
Rename the columns as follows:

- Date
  - Materials
  - Printing
  - Assembly
  - Total Production Cost
  - Shipping
  - Total Cost
4. Verify that the data type is correct for each column by looking at the icon next to the column names. For example, the Date column uses the **Date/Time** data type, so 12:00:00 AM is added to each entry. Right-click the column name and select **Change Type > Date** so that only the date appears.

The screenshot shows the Microsoft Power Query Editor interface. On the left, the 'Queries [1]' pane lists a single query named 'cr71f\_costtrends'. In the main area, there is a table with columns labeled 'Date' and 'Assembly'. The 'Date' column contains dates from 1/30/2011 to 7/30/2011. The 'Assembly' column contains numerical values: 8808, 8174, 8634, 8048, 8728, 8934, and 8598. A context menu is open over the first row of the 'Date' column. The menu path 'Date' -> 'Change Type' is highlighted. The 'Change Type' submenu is open, showing various options: Decimal Number, Fixed decimal number, Whole Number, Percentage, Date/Time (which is checked), Date, Time, Date/Time/Timezone, Duration, Text, True/False, Binary, and Using Locale... .

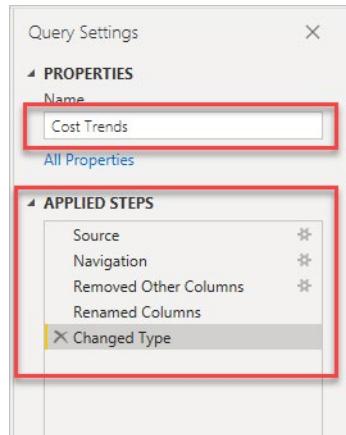
Common data types and their icons are shown in the following table.

| Icon | Data type |
|------|-----------|
|      | Date/Time |

| Icon                                                                                | Data type    |
|-------------------------------------------------------------------------------------|--------------|
|    | Date         |
|   | Whole number |
|  | Percentage   |

| Icon                                                                              | Data type |
|-----------------------------------------------------------------------------------|-----------|
|  | String    |

5. In the **Query Settings** tab, rename the table by selecting **Name** and entering **Cost Trends**. All transformation actions are listed under **Applied Steps**. You can undo a step by deleting it from the list.



6. Select **Close & Apply** from the **Home** ribbon to load the data with the transformations into Power BI Desktop.

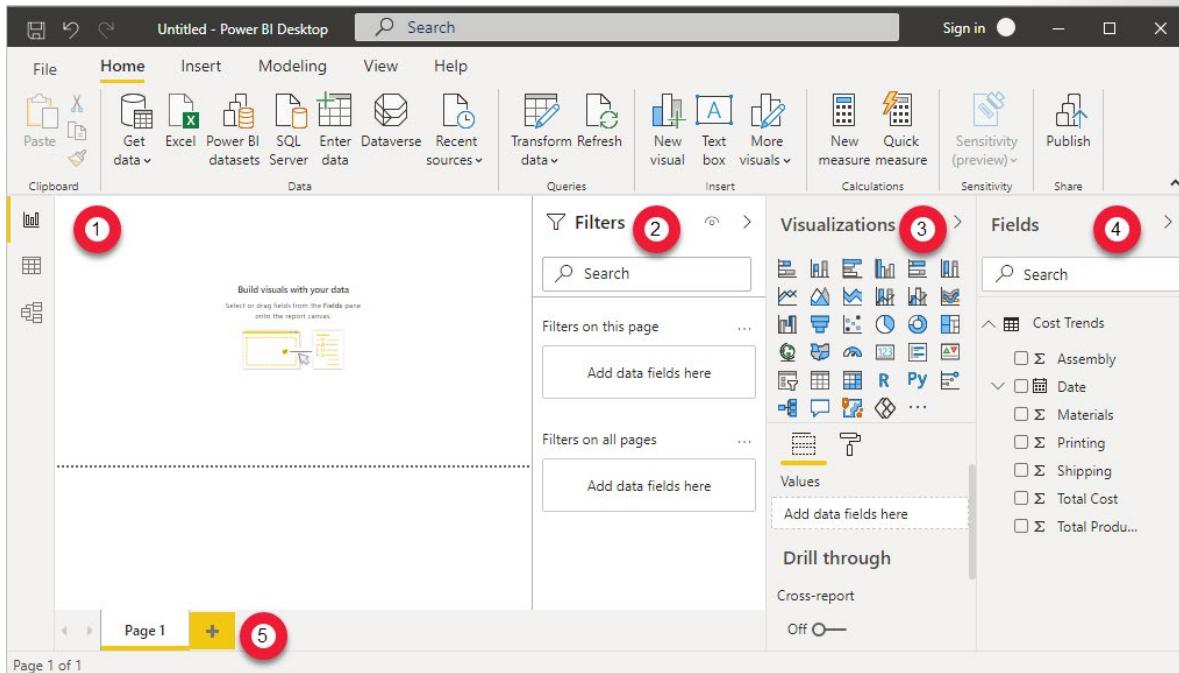
For more information about Power Query Editor, see the Summary section at the end of the module.

## Create a Power BI report

Previously, you connected to your Dataverse for Teams table data and then transformed it. Now, you can build a report with interactive visualizations by using the **Report** view in Power BI, which consists of the following areas:

- **Report canvas** - Where visualizations are created and arranged (number **1** in the image).
- **Filters pane** - This allows you to filter and only work with subsets of data (number **2** in the image).
- **Visualizations pane** - Where you can select and edit the visualizations (number **3** in the image).
- **Fields pane** - Where you can choose the data to include in the visualizations or include it as a filter (number **4** in the image).

- **Pages tab** - Where you can create more report canvases (number 5 in the image).



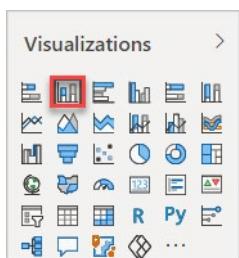
The basic steps to add visualizations to the report are:

1. Select a visualization from the **Visualizations** pane.
2. Select and drag the required fields from the **Fields** pane into the **Visualizations** pane or **Filters** pane.
3. Move and resize the visualization in the report canvas.
4. Make more edits to the visualization in the **Visualizations** pane.

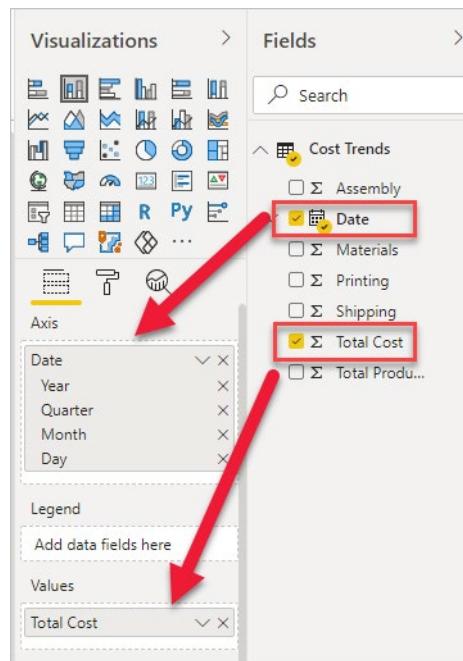
## Add a stacked column chart

To add a stacked column chart, follow these steps:

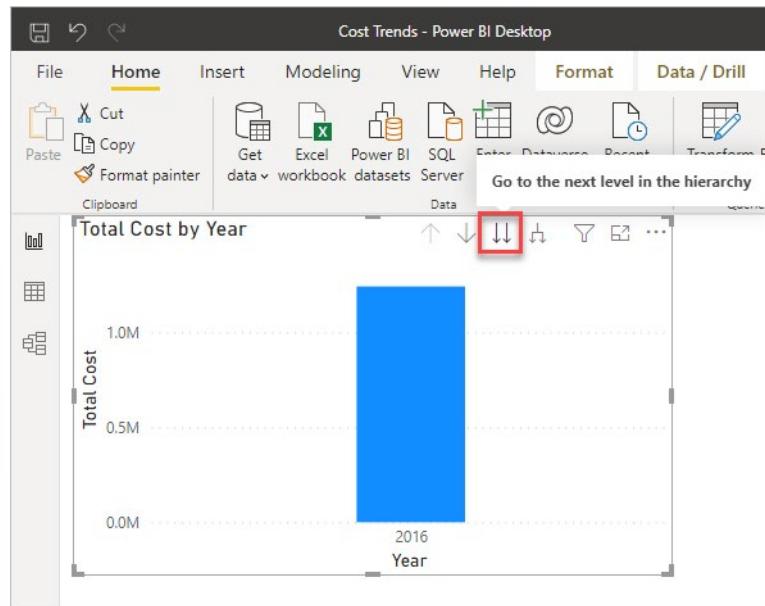
1. Select the **Stacked Column Chart** icon from the **Visualizations** pane.



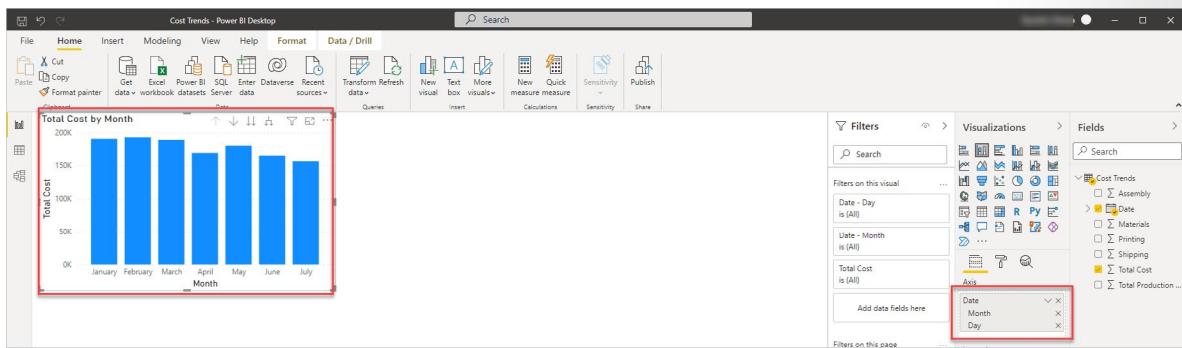
2. Select and drag the following elements from the **Fields** pane:
  - **Date** field to **Axis**
  - **Total Cost** field to **Values**



3. Notice that the chart is showing the data by years. You can drill in to see the data by quarter and month by selecting the **Go to the next level in the hierarchy** icon. In the next step, you will edit the chart to show the data by month.



4. With the chart selected in the report canvas, remove **Year** and **Quarter** under **Axis** so that the graph shows the data by month.



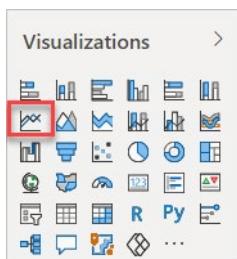
5. Resize the chart in the report canvas by selecting and dragging the indicated border controls. Have the chart fill up the upper-left portion of the canvas.



## Add a line chart

To add a line chart, follow these steps:

1. Select the **Line Chart** icon in the **Visualizations** pane.



Note: If you already have a visualization selected in the report canvas, then selecting another visualization will replace it. Make sure that you don't have a visualization selected by selecting a blank area of the report canvas before you create a new one.

2. Select and drag the following entities from the **Fields** pane.

**For Axis:**

- Date (remove Year and Quarter)

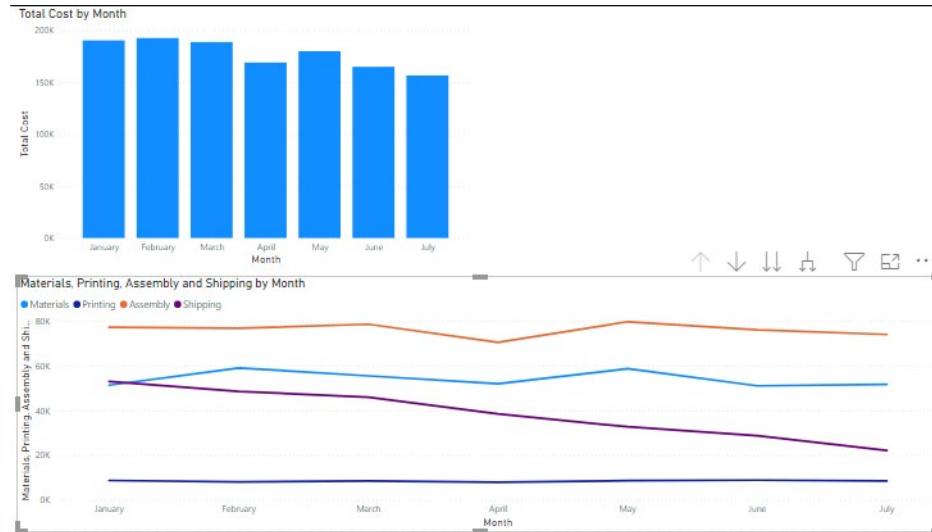
| Axis  |     |
|-------|-----|
| Date  | ✓ X |
| Month | X   |
| Day   | X   |

**For Values:**

- Materials
- Printing
- Assembly
- Shipping

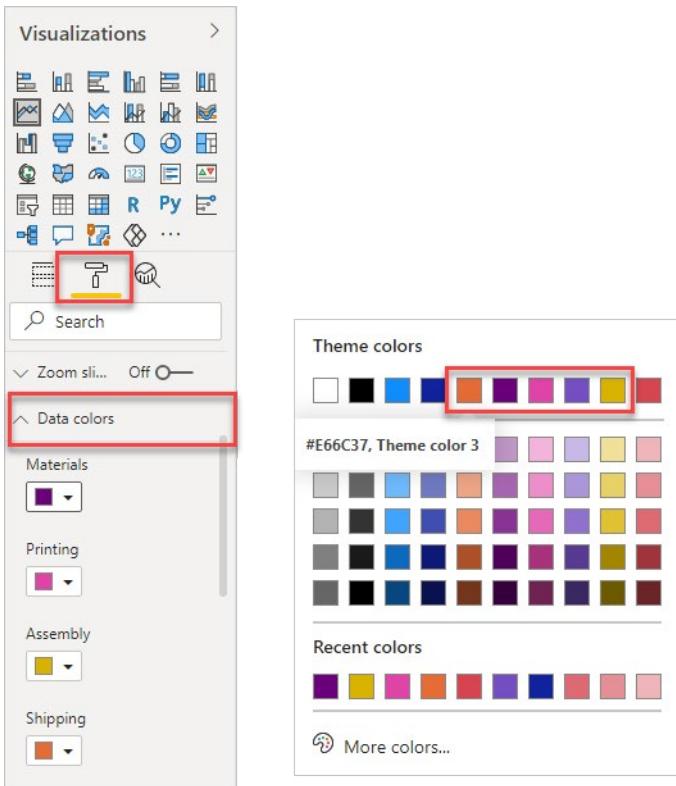
| Values    |     |
|-----------|-----|
| Materials | ✓ X |
| Printing  | ✓ X |
| Assembly  | ✓ X |
| Shipping  | ✓ X |

3. Select and drag to resize the line chart to cover the lower half of the report canvas.



4. Change the data colors by going to the **Visualizations** pane, selecting the **Format** tab, and then expanding **Data colors**. Use the drop-down list next to each color and select the following theme colors:

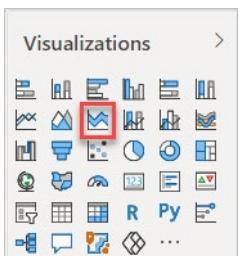
- **Materials** - Theme color 4
- **Printing** - Theme color 5
- **Assembly** - Theme color 6
- **Shipping** - Theme color 3



## Add a stacked area chart

To add a stacked area chart, follow these steps:

1. Select the **Stacked Area Chart** icon in the **Visualizations** pane.



2. Select and drag the following entities from the **Fields** pane to the **Values** area.

**For Axis:**

- Date (Remove Year and Quarter)

| Axis  |   |   |
|-------|---|---|
| Date  | ▼ | X |
| Month | ▼ | X |
| Day   | ▼ | X |

**For Values:**

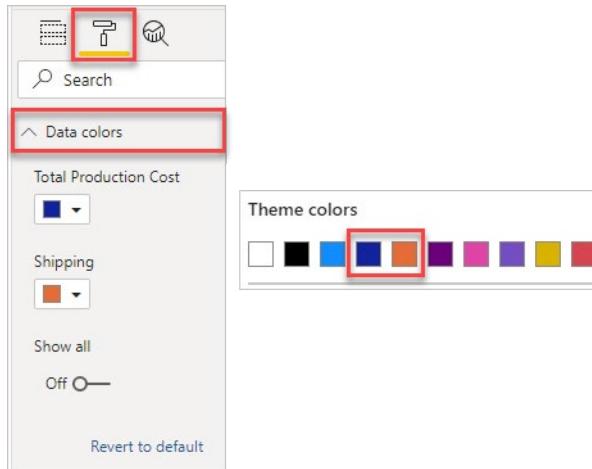
- Total Production Cost

- Shipping

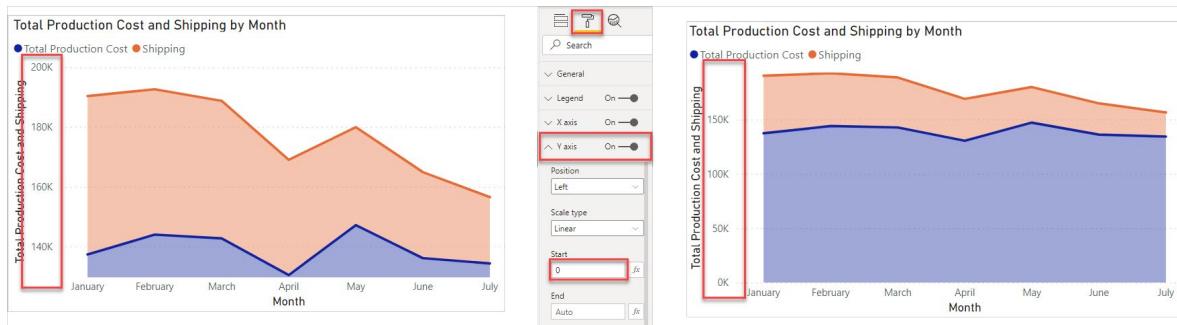


3. Change the colors of the area chart to make it consistent with the rest of the report. In the **Visualizations** pane, select the **Format** tab and then expand **Data colors**. Use the drop-down list next to each color and select the following theme colors:

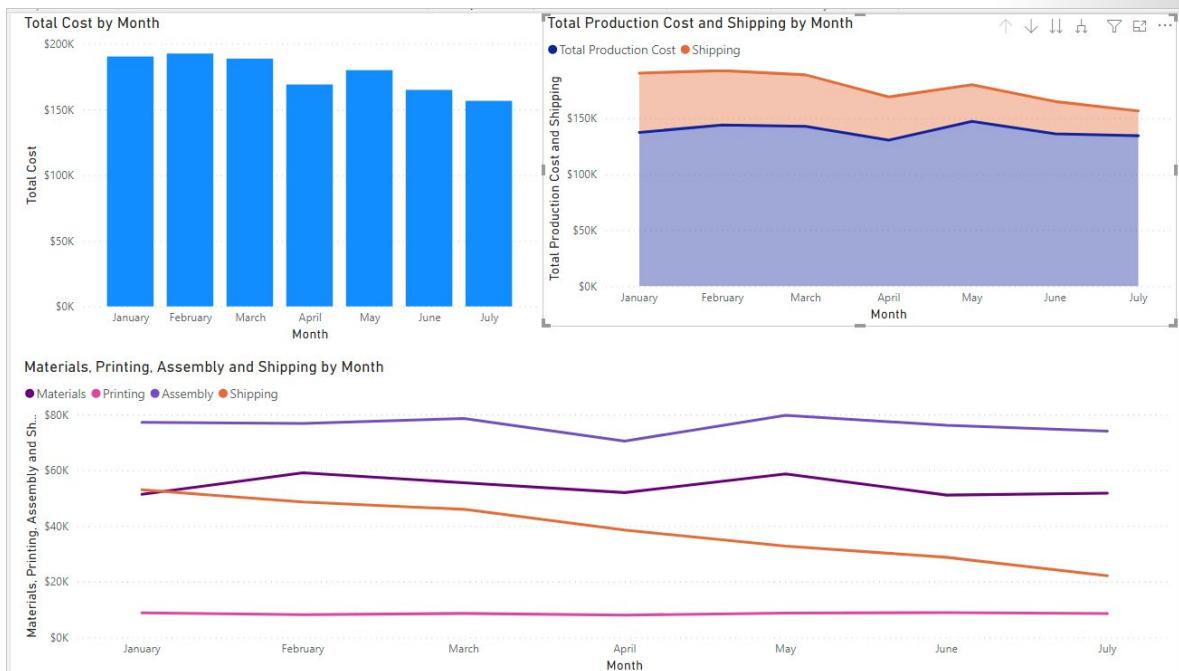
- Total Production Cost** - Theme color 2
- Shipping** - Theme color 3



4. If the new chart has automatically set the Y axis to zoom in on the shipping data, in the **Format** tab expand the **Y Axis** section and type in **0** for the start. The graph now shows both the Total Production Cost and the Shipping information.

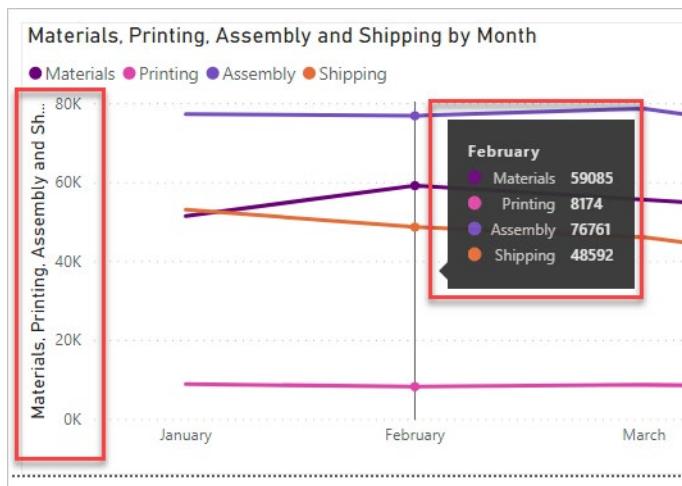


5. Select and resize the chart in the report canvas to occupy the existing space.



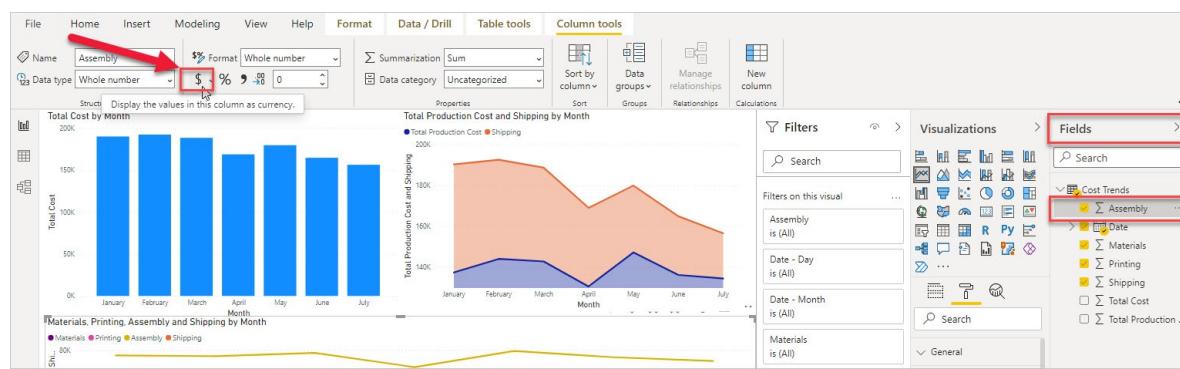
## Change data to use currency format

Notice that the report shows data in the legends and pop-out windows as simple numbers without formatting. It would be more useful to have the data appear as currency.



To change to currency format, follow these steps:

- Under the **Fields** pane, select the **Assembly** field.
- In the **Column tools** ribbon, select the **Currency** format icon.



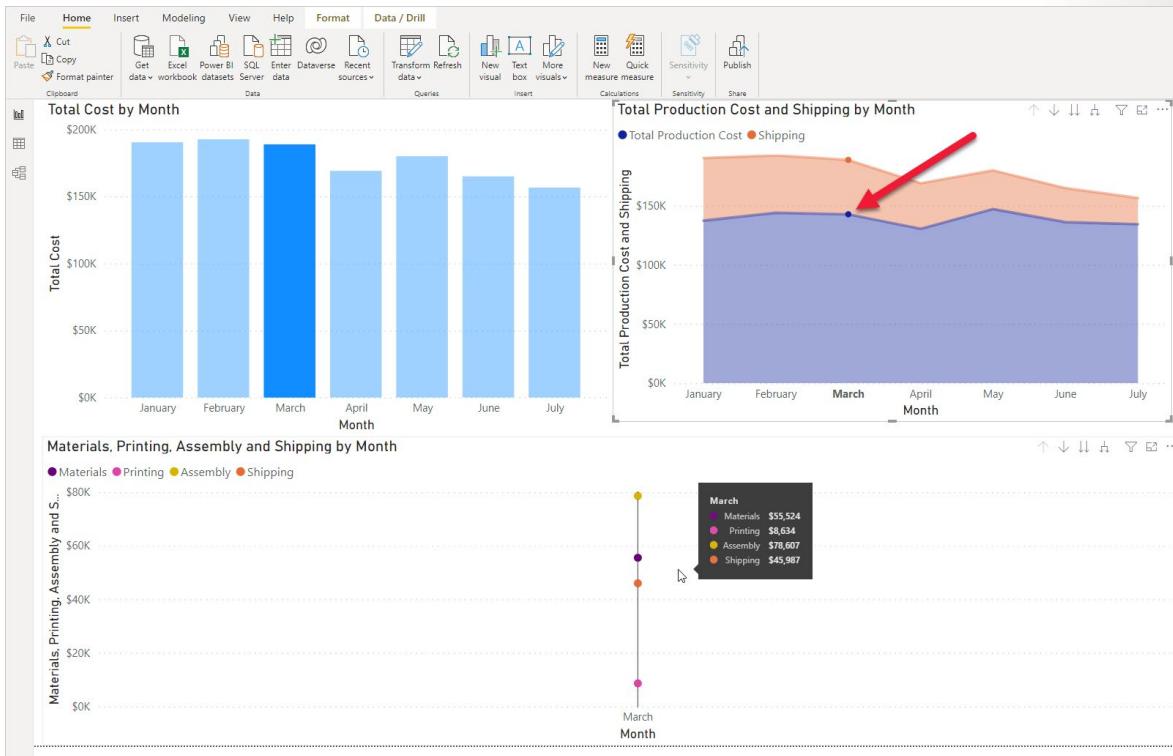
### 3. Repeat these steps for **Materials, Printing, Assembly, and Shipping**, **Total Cost**, and **Total Production Cost**.

After you have completed these steps, your data will appear as currency.



## Explore the report

Power BI links your charts to be interactive when selecting a visual in one chart; the other charts will automatically adjust. For example, if you click on the data point for March under **Total Production Cost and Shipping by Month**, the other two charts will adjust to show data for that month. **Total Cost by Month** will highlight the data for March while the **Materials, Printing, Assembly, and Shipping by Month** line graph shows just the data point for March. Click on the same data point to revert all the charts back to normal.

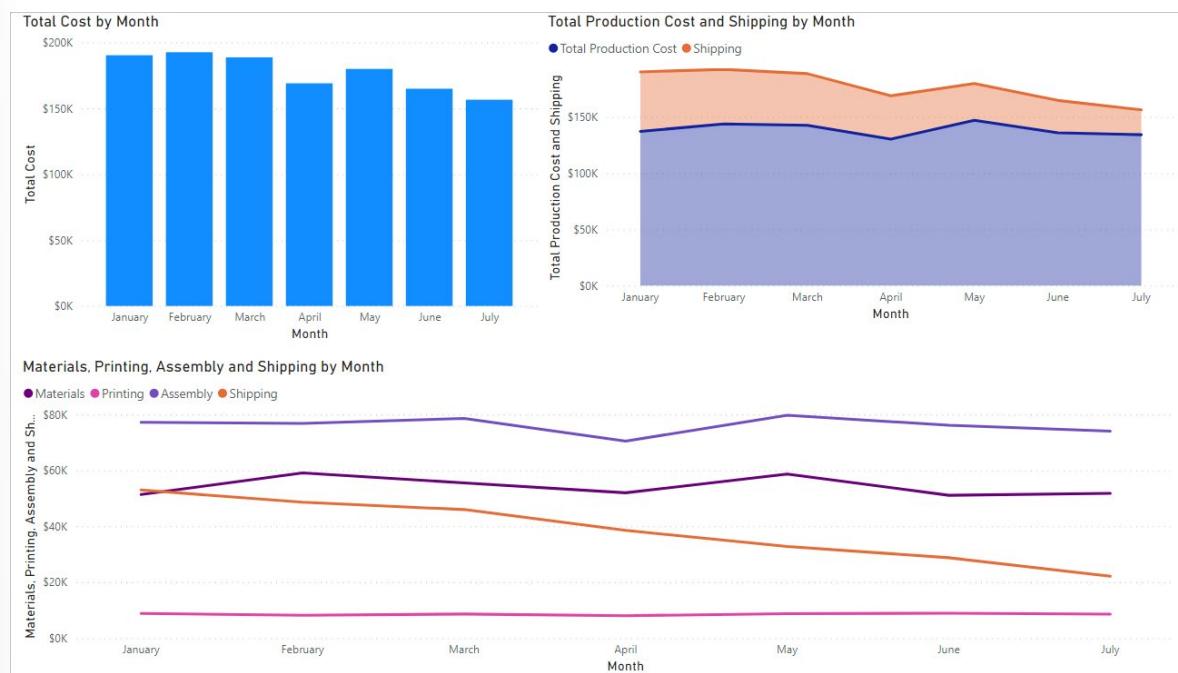


Experiment by selecting other visuals in the other charts. As you add additional tables into your reports, you will create more complex reports that automatically interact to help you and your users explore the data. For more information on Power BI visual interactions and how to configure them, please see the links that are provided in the Summary section at the end of the module.

## Save the report

Save the report by selecting **File > Save as** in the menu. Enter **Contoso Cost Trends** as a name and then select **Save**. Power BI Desktop uses a **.pbix** file extension for its saved files.

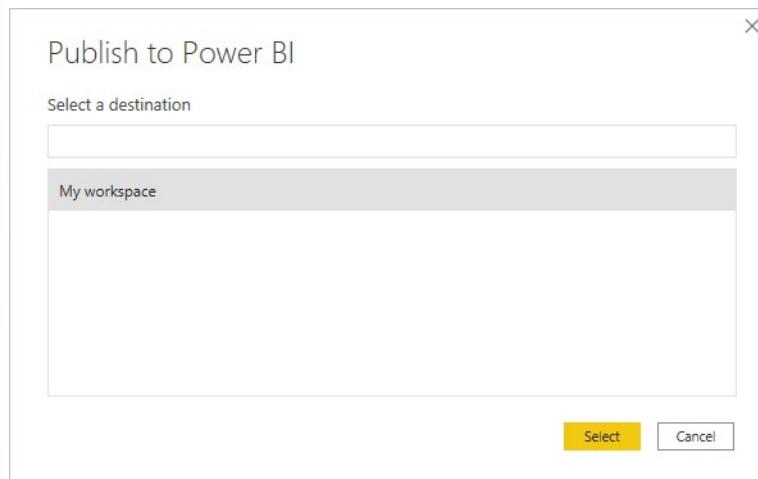
The final report should look similar to the following image.



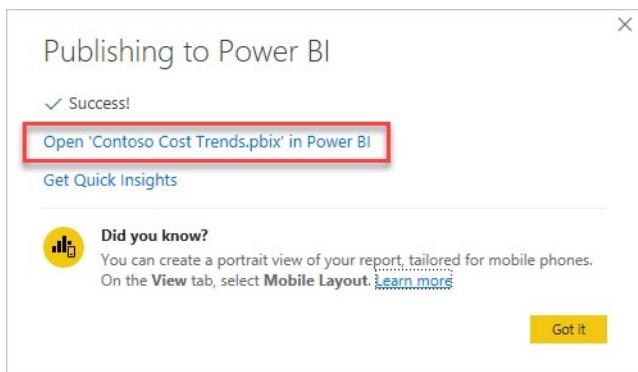
## Publish the report

For users to access the report in Teams, the report and dataset need to be published to a workspace. A Power BI **workspace** is an area to store reports, workbooks, datasets, and dashboards in Power BI service. Every Power BI user has a **My workspace** area that only they have access to. From **My workspace**, users can share reports and dashboards. Members of workspaces that are created in Power BI have access to any reports that are published to them. For more information about Power BI workspaces, see the links in the Summary unit at the end of the module.

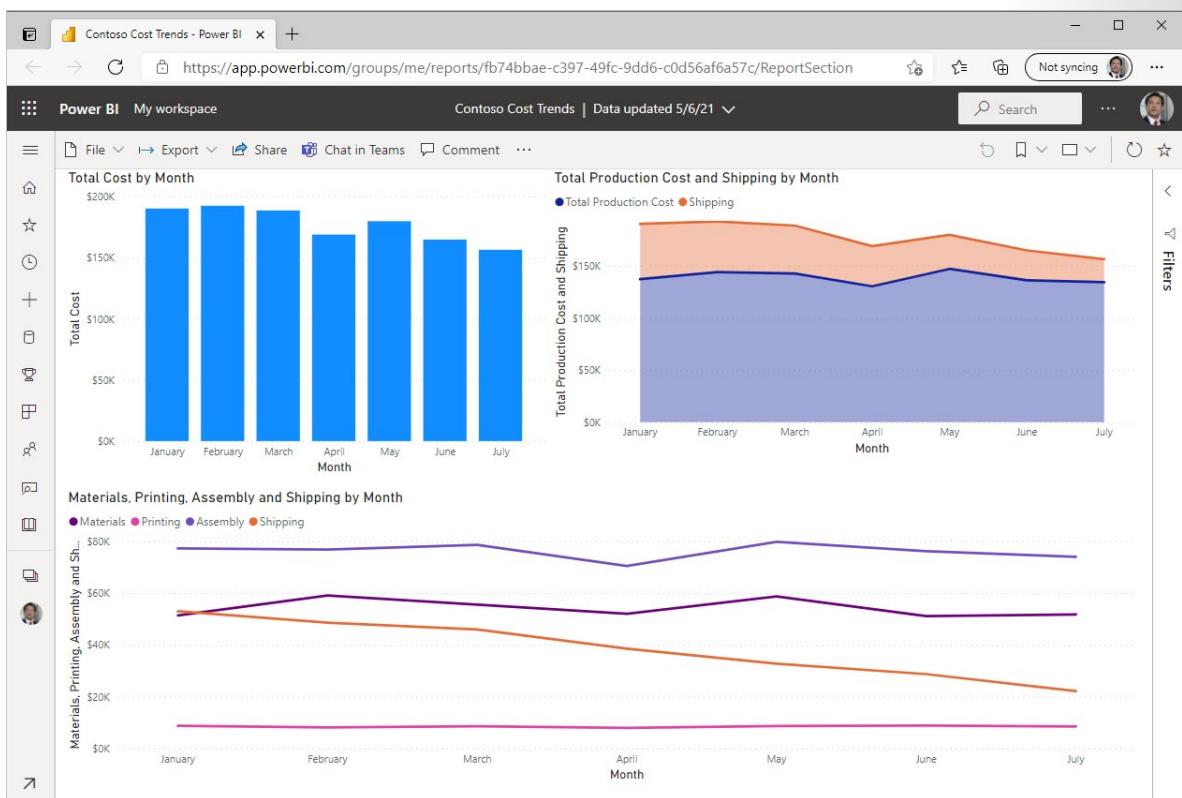
1. Open the report in Power BI Desktop.
2. From the **Report** view in Power BI, select **Publish** from the **Home** ribbon.  
If you haven't signed in to Power BI Desktop yet, you will be prompted for credentials.
3. After you have successfully signed in, select a Power BI workspace to publish the report.



4. A **Publishing to Power BI** pop-up window appears. When the publishing process has completed, select **Open 'Contoso Cost Trends.pbix' in Power BI**.



5. The default web browser connects to the Power BI website with the published report.



## Share the data

Power BI reports are designed so that people can view and interact with them to gain insights and make better decisions. The Power BI for Teams app allows you to collaborate with your team members on data and take action.

In this unit, you will learn how to:

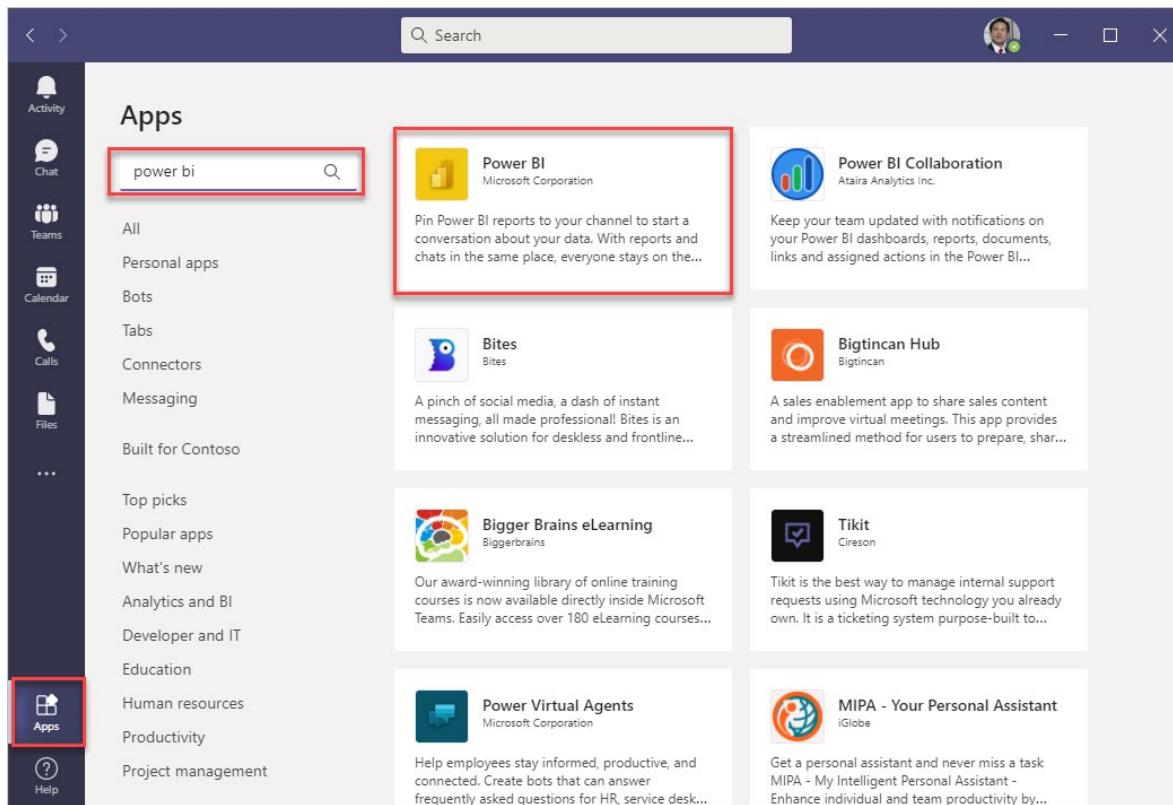
- **Add the Power BI for Teams app** so that you can access your Power BI data without leaving Teams.
- **Grant permissions to reports** to allow your coworkers and team members access.
- Use the Power BI tab for Microsoft Teams to **embed interactive reports in channel and chat tabs**.

- **Start a chat in Teams** to start a conversation while you are viewing the report.
- **Reference a report in the Teams message box** when you are collaborating with coworkers.
- **Add a comment to a report** to start a conversation with colleagues.

## Add the Power BI for Teams app

To add the Power BI for Teams app, follow these steps:

1. Launch Teams or use your browser to open <https://teams.microsoft.com><sup>25</sup>.
2. In Teams, select **Apps** in the lower-left corner of the screen.
3. In the search box, enter **Power BI** and then select it from the search results.



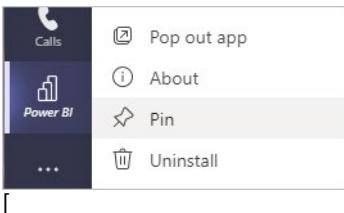
4. Select **Add** to install Power BI.

If you've never signed in to Power BI before, sign in and then go through the new user setup prompts.

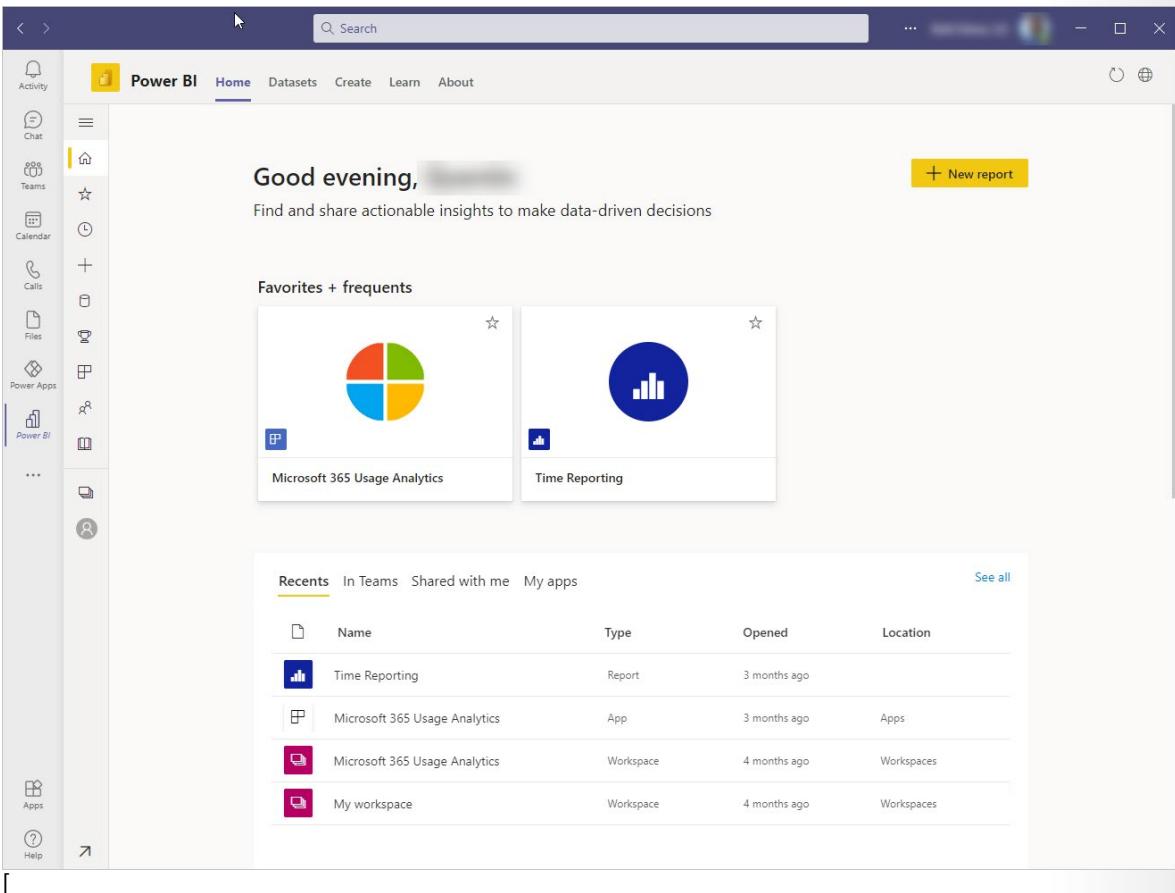
[!NOTE]

Pin the Power BI app to the app launcher to quickly access it at any time. Right-click the Power BI icon and select **Pin**. You can also run the app as a separate window from Teams by selecting **Pop out app**.

<sup>25</sup> <https://teams.microsoft.com/?azure-portal=true>



The Power BI app for Teams launch page gives you quick access to your favorite reports, dashboards, and workspaces. Also, you can explore Power BI training and recommended apps.



## Share a report

Reports that are published to Power BI workspaces automatically grant access to everyone who is a member of the workspace. If you published the report to **My workspace**, only you have access to the report by default.

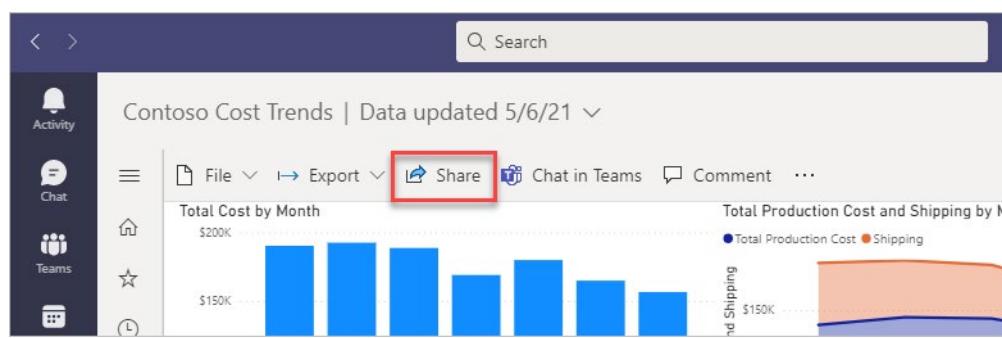
[!NOTE]

When a user tries to access a Power BI report that they don't have access to, they are prompted through a request access dialog box to provide a note that explains their reason for access. The owner of the report will receive an email to review the request and approve if appropriate.

To create a share link, follow these steps:

1. Open the report in the Power BI for Teams app.
2. Set any filters or slicers that you want the recipients to see. The share link will contain this information and users will see the filtered report when using the shared link.

3. Select **Share** from the menu bar.

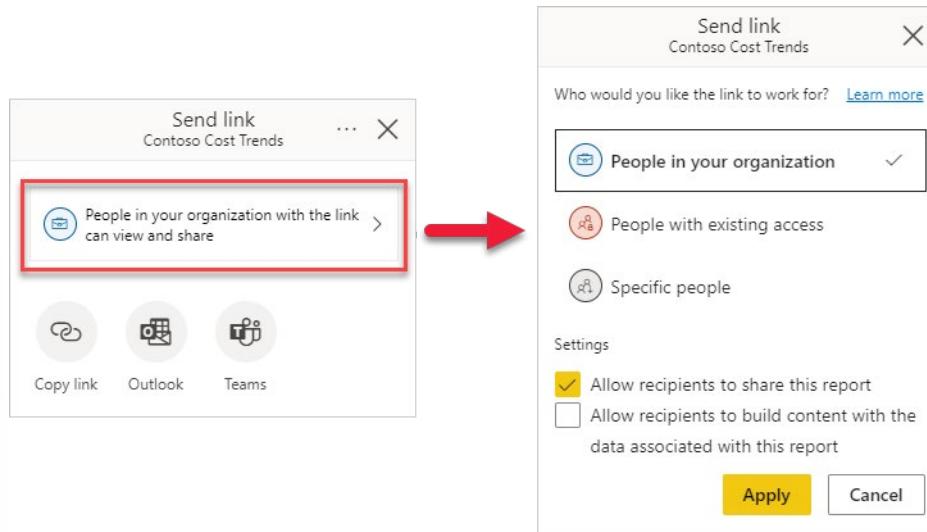


4. Expand the dialog box to define who can use the link. Choose the appropriate settings and then select **Apply**.

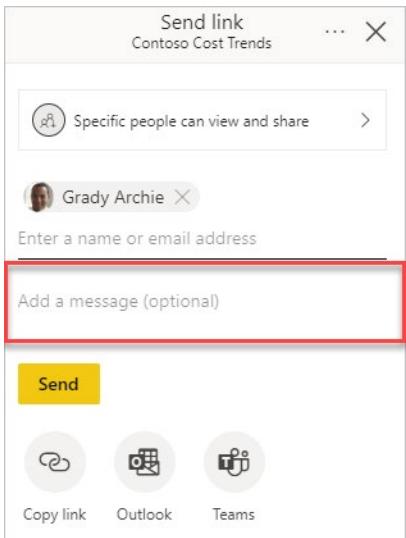
- **People in your organization** - This link grants access to people in your organization. It doesn't work for external users.
- **People with existing access** - Use this option when you want to generate a link and not change the existing permissions.
- **Specific people** - This link prompts you to provide the name of specific people, groups, or external guests in your organization's Microsoft Azure Active Directory (Azure AD). You can't grant access to external users that aren't guests of your organization.

[!NOTE]

Use the **Specific people** option to ensure everyone has access when sharing as a tab in a channel, chat, or meeting.

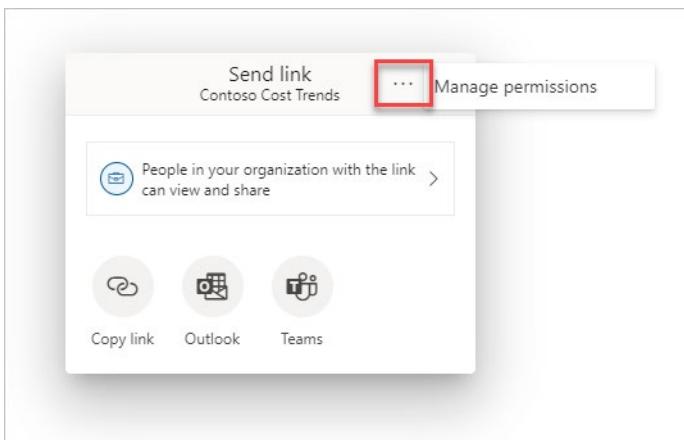


5. Select whether you want to **Copy link** to the clipboard, open an **Outlook** email with the link, or post the link to a **Teams** channel. If you share with **Specific people**, you can enter a message and **Send** an email immediately.

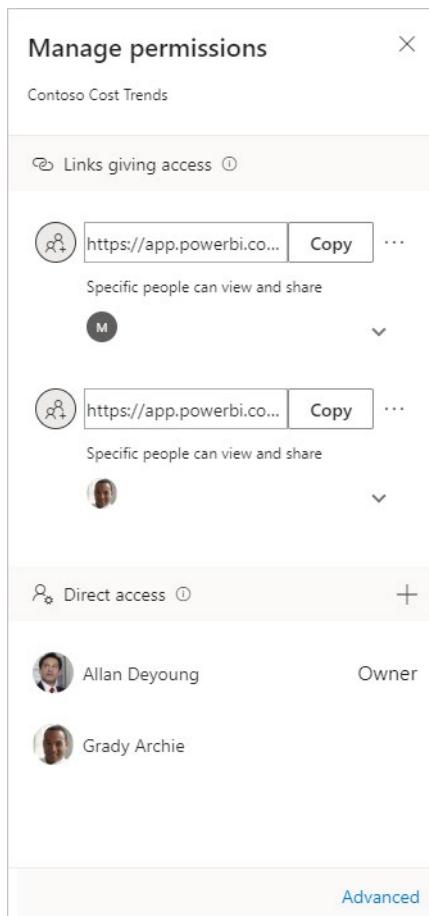


To manage report permissions, follow these steps:

1. Open the report in Power BI for Teams.
2. Select **Share** from the menu bar.
3. Select the **More options** ellipsis (...) in the **Send link** windows title and then select **Manage permissions**.



4. From the **Managed permissions** panel, you can review **Links giving access** and who has **Direct access** permissions to the report.

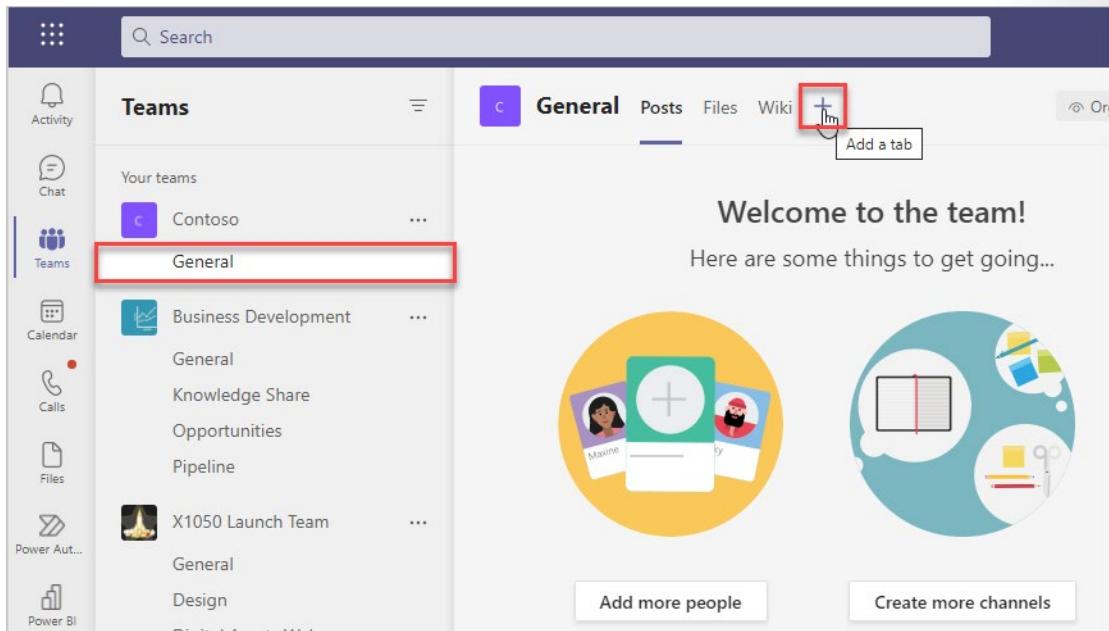


## Embed interactive reports in channel and chat tabs

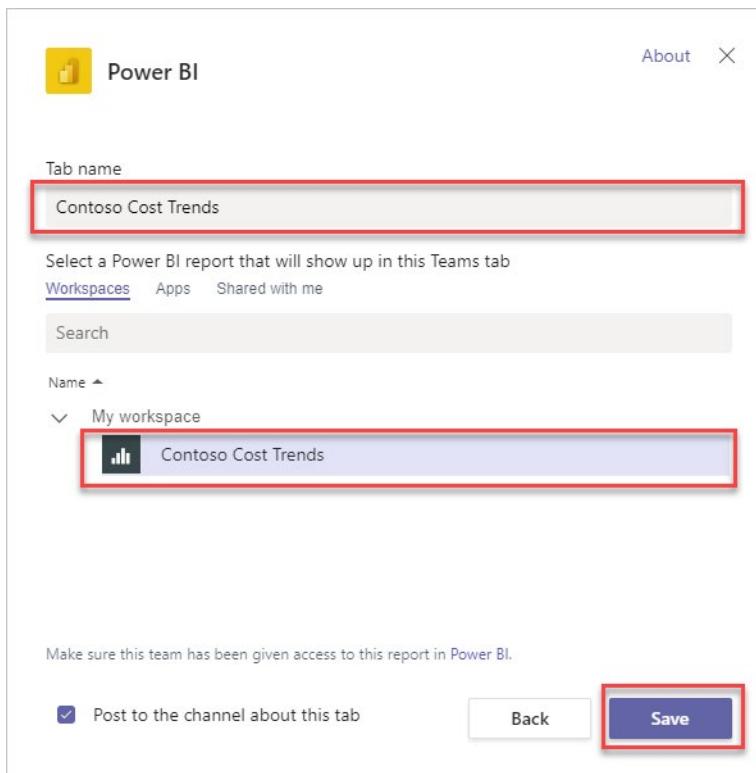
Adding a Power BI tab in Teams helps make it easier for your team members to find the report. They can view and interact with the report without leaving the channel or chat.

1. Open the Teams app or use your browser to open <https://teams.microsoft.com><sup>26</sup>.
2. Open a chat or channel to add the Power BI report.
3. Select the **Add a tab** icon (+).

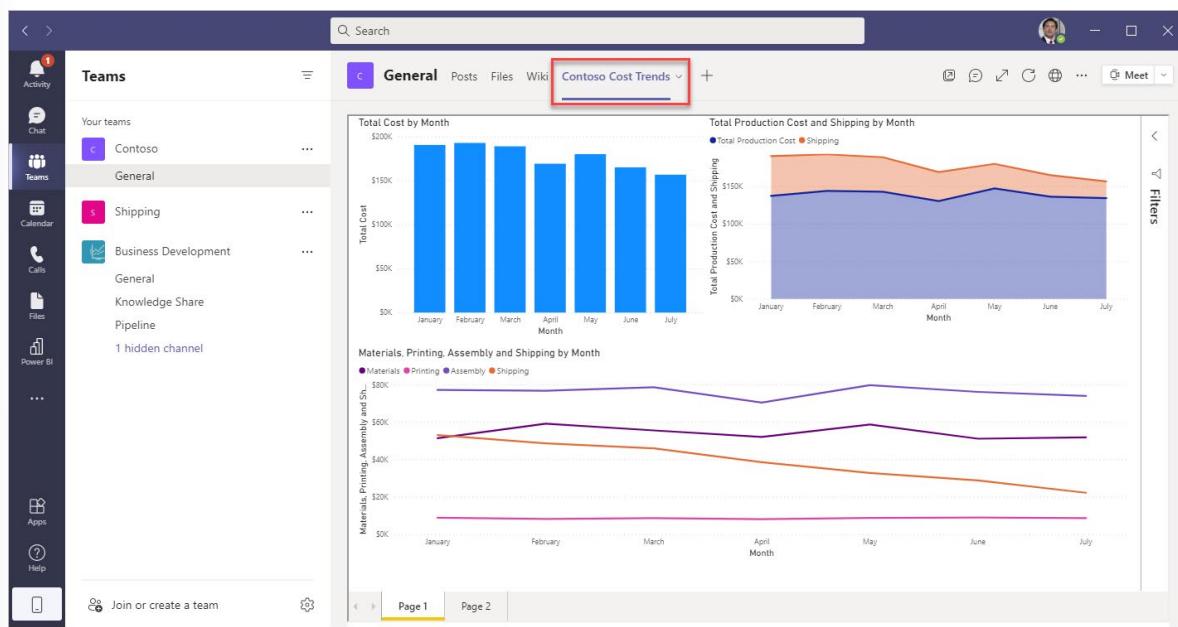
<sup>26</sup> <https://teams.microsoft.com/?azure-portal=true>



4. Select the **Power BI** app.
5. Enter a **Tab name** and then select the report from the workspace section. Select **Save** to continue.



The report will appear as a tab in the channel or chat.

**[!NOTE]**

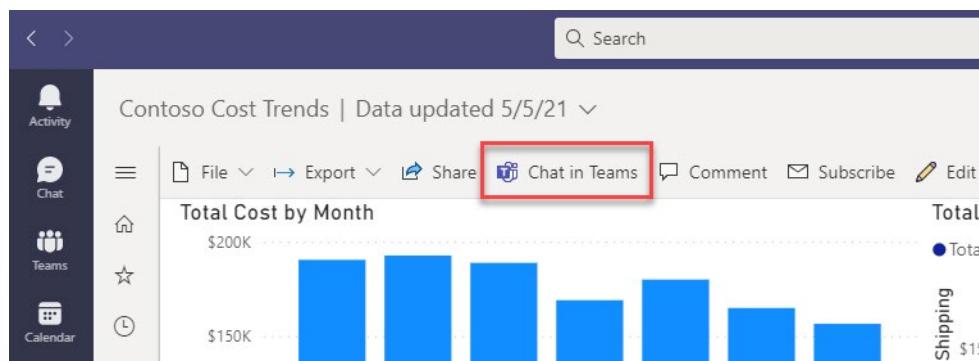
Adding a Power BI tab does not change permissions, and only users that have access can view the report. Users who don't have access will be prompted to request access when they select the tab.

## Start a chat in Teams

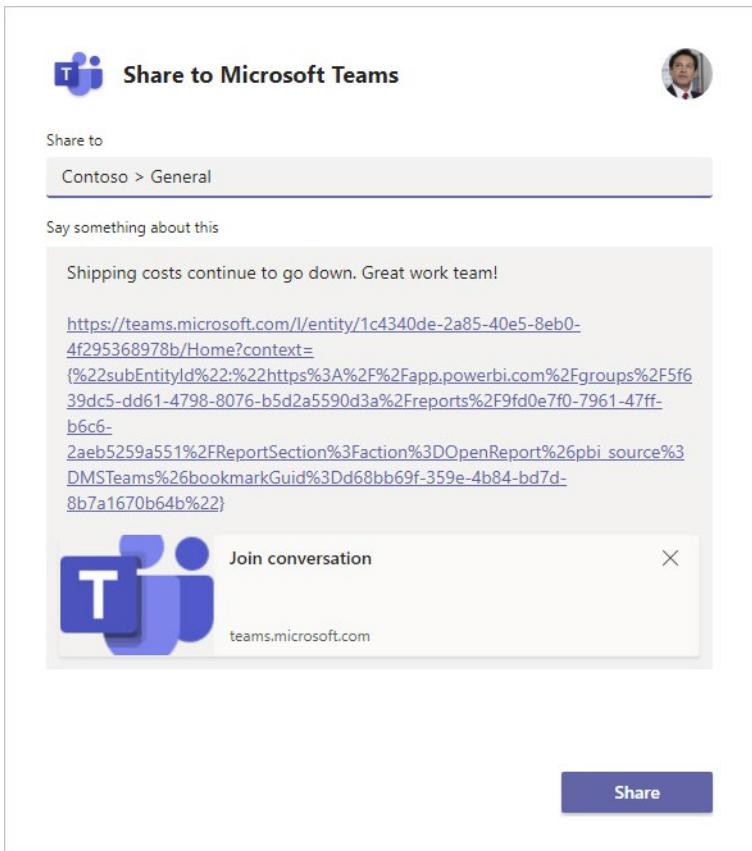
You can start a conversation in a Teams channel when you are viewing reports in the Power BI for Teams app. The conversation can be about the report or you can direct it to an individual visual.

To start a chat in Teams about the report, follow these steps:

1. Open the report in the Power BI for Teams app.
2. Select **Chat in Teams**.

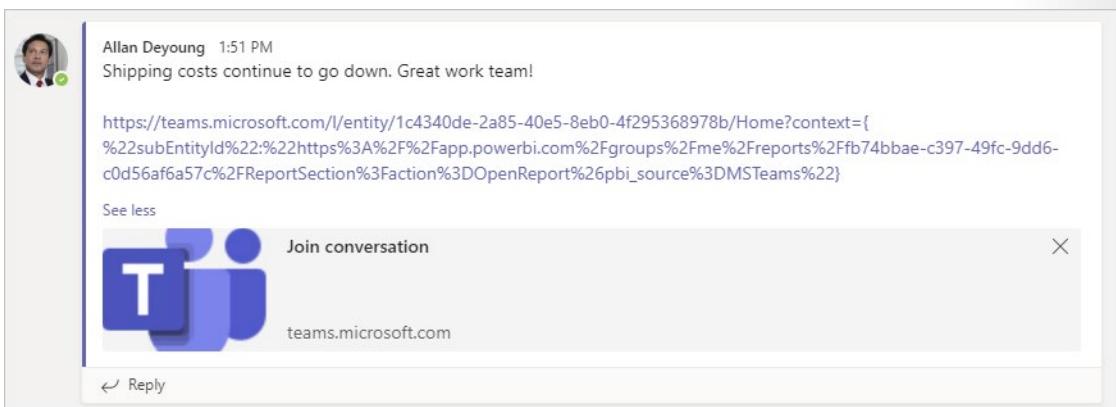


3. If prompted, fill out the authentication prompts for Teams.
4. Enter the name of a team or channel in the **Share to** field. Enter a message without overwriting or changing the URL.



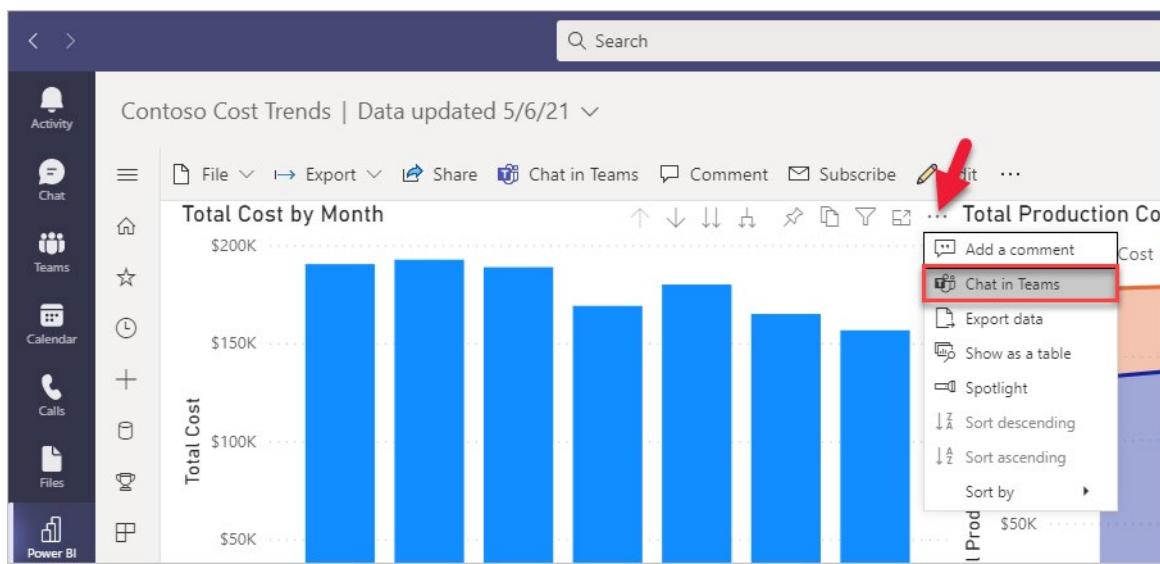
5. Select **Share** when you are finished.

In the channel, a new conversation will be started with a link to the report.

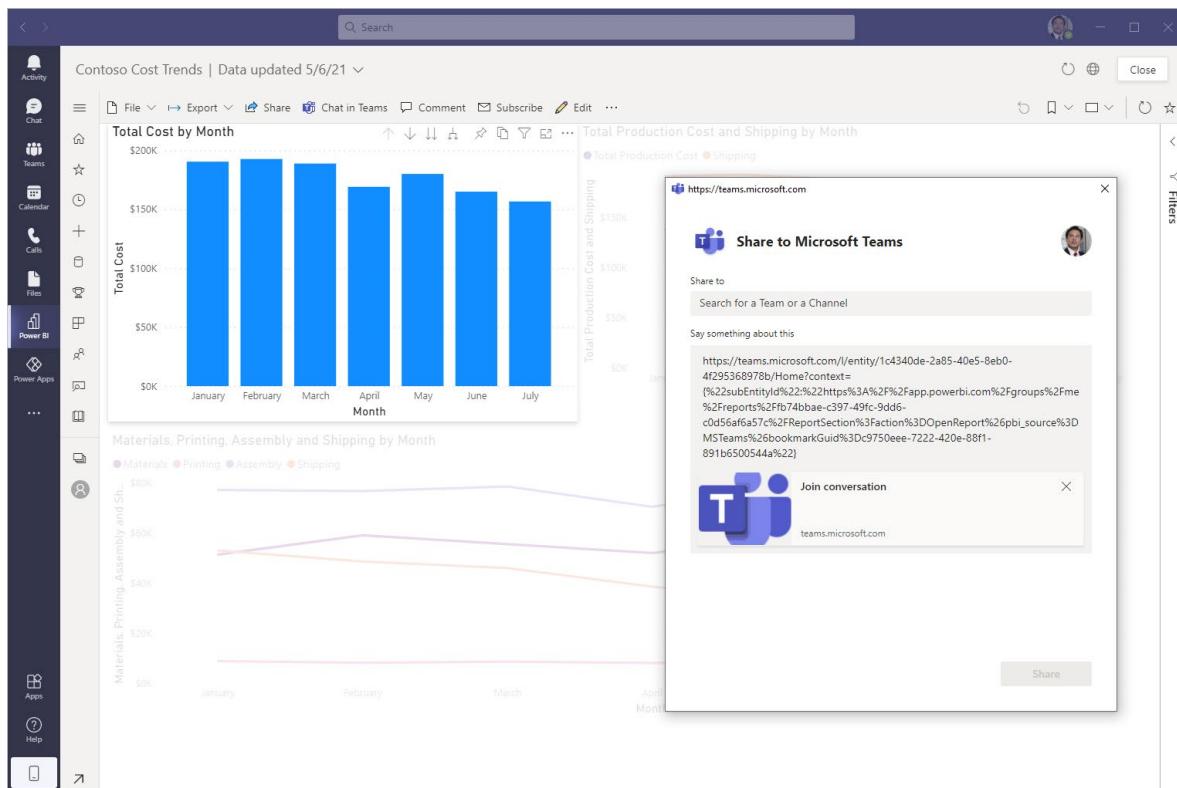


To start a chat in Teams about an individual visual, follow these steps:

1. Open the report in the Power BI for Teams app.
2. On the report visual, select the **More options** ellipsis (...) and then select **Add a comment**.



3. If prompted, fill out the authentication prompts for Teams.
4. Enter the name of a team or channel in the **Share to** field. Enter a message without overwriting or changing the URL. Select **Share** when you are finished.



When users select the link in Teams, the selected visual will be highlighted.

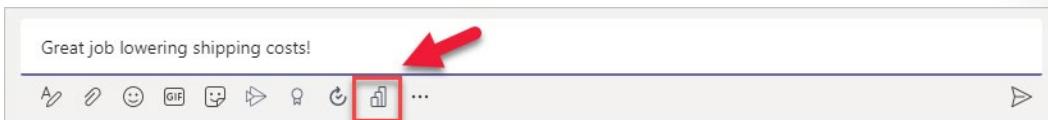
[!NOTE]

**Chat in Teams** starts a Microsoft Team conversation with a link to the report. It does not update the permissions, and only users with permissions to the report can access it. Users who don't have access will be prompted to request access when they select the link.

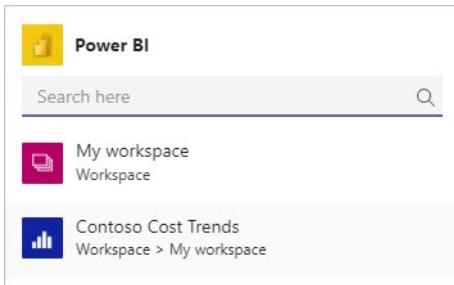
## Reference a report in the Teams message box

To reference a Power BI report from a channel or chat, follow these steps:

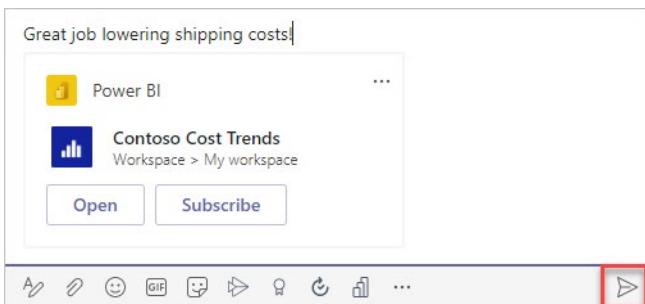
- From a Teams message box, select the **Power BI** icon.



- Search for and select the report to include in the chat.



- Send the message when you are done.



Users can select to **Open** the Power BI report, or they can select **Subscribe** to set up a schedule to receive email updates. For more information about subscribing to a Power BI report, see the links in the Summary unit at the end of the module.

## Share Power BI during a meeting

Data-driven decision-making allows you and your team to spot trends, drive actions, and improve outcomes. Review data each week with your team to enable better data-driven decisions. Add Power BI reports to meetings by adding a link to the invite, adding the tab in a channel, sharing the link in the meeting chat, or sharing your screen.

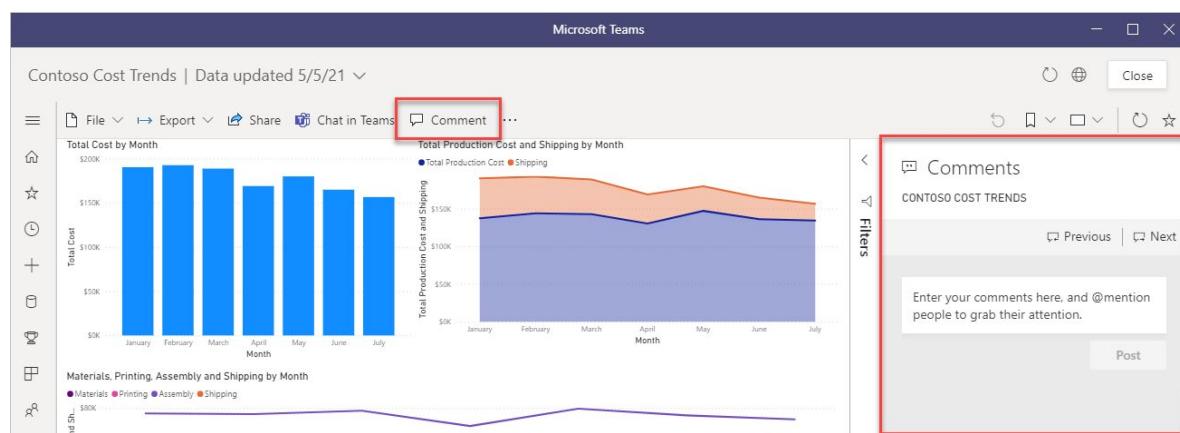
## Add comments to a report

You can add comments to a report or even on individual visuals on the report. To target comments to individuals, use the **@mention** function to grab their attention. Report filters or slicer settings will be captured with the comment so that you can view the report as it was when you first commented.

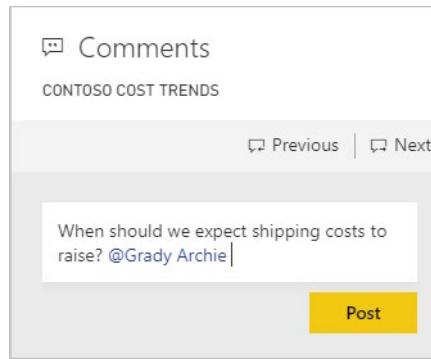
To create a general comment on the report, follow these steps:

- Open the report in the Power BI for Teams app.

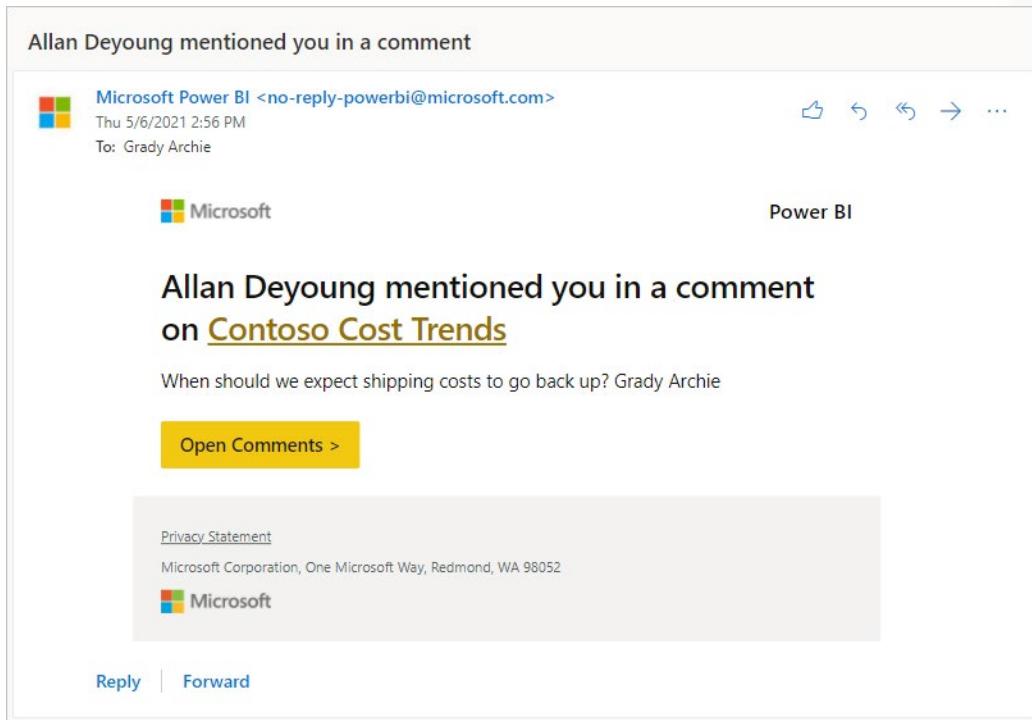
2. Select **Comment** to open the **Comments** pane.



3. Use **@mention** to alert others to your comment so that they can respond. Select **Post** when you are finished.



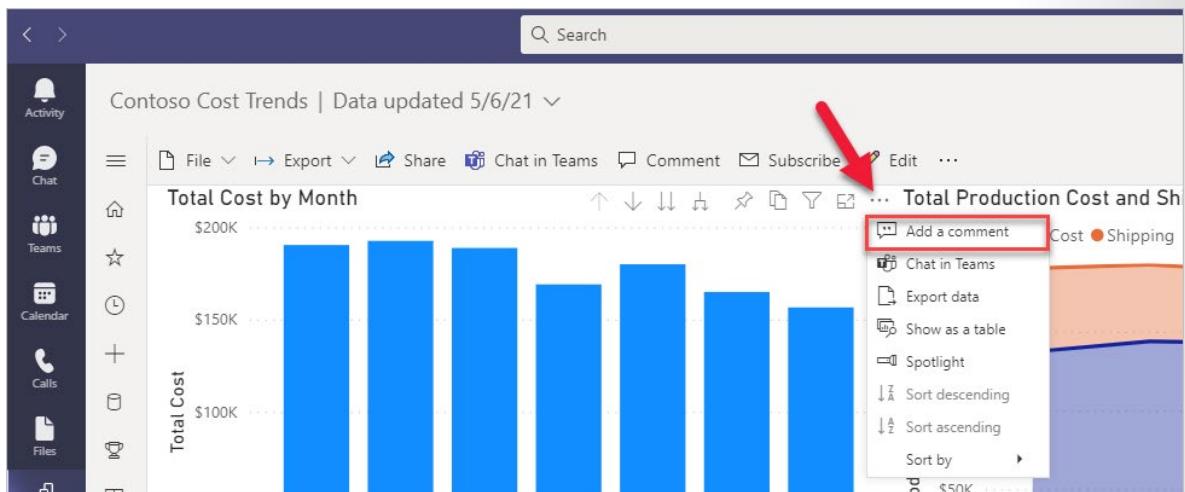
An email will be sent to the person who is tagged in the **@mention** function.



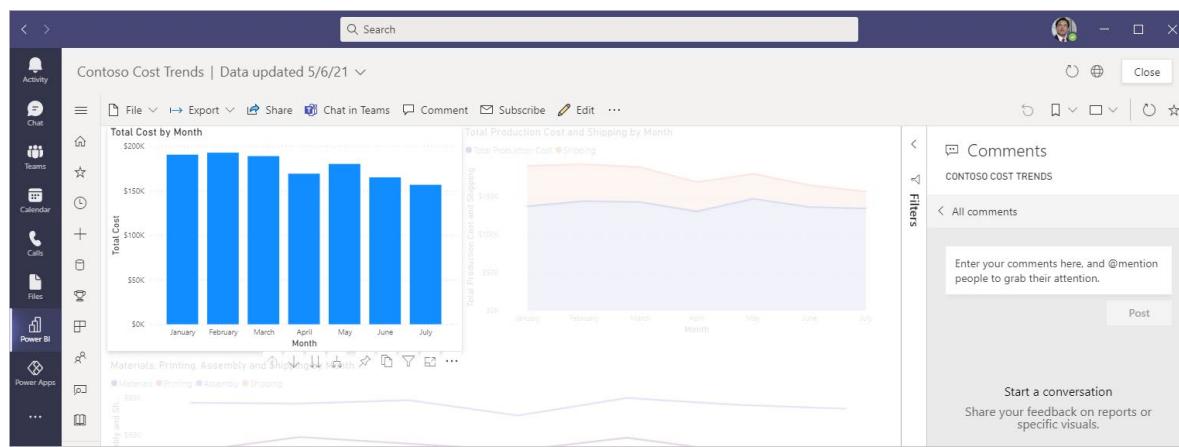
When the person selects **Open Comments**, a browser window will open to the report on the Power BI website, where they can reply to the comment.

To comment on an individual visual, follow these steps:

1. Open the report in the Power BI for Teams app.
2. On the report visual, select the **More options** ellipsis (...) and then select **Add a comment**.



3. The selected report visuals will be highlighted when you enter a comment. Use **@mention** functions in the same way that you would for general comments.



## Summary

In this module, you learned how to connect and transform Dataverse for Teams data into Power BI. You also learned to use that data to build and edit Power BI reports with visualizations. You also learned how to share those reports with others and access them without leaving Microsoft Teams.

Empower your users with Power BI reports by using Dataverse for Teams data so that they can visualize, interact, and share insights with others across the organization. With Power BI and Dataverse for Teams, you can make better decisions, accelerate business transformation, and grow the organization.

## Links related to modules for specific topics

For more information, see the following links:

- Provision a Dataverse for Teams Environment and create a table
- Power BI Pro licensing
- Power Query Editor
- Power BI Visual Interactions
- Subscribe to a Power BI Report
- Power BI Workspaces

# Module 6 AI Builder and Power Virtual Agents

## Get started with AI Builder

### Introduction to AI Builder

Welcome to AI Builder. This self-paced module will help you build an AI model from the beginning and will show you how to use it in your business without writing a single line of code.

### AI Builder defined

AI Builder is a Microsoft Power Platform capability that helps you improve your business performance by automating processes and predicting outcomes. By using AI Builder, you can quickly bring AI to your apps and flows that connect to your business data that is stored in the underlying data platform (Microsoft Dataverse) or in various cloud data sources, such as SharePoint, OneDrive, or Azure.

AI models that you create in AI Builder can help provide intelligence to enhance your business. AI Builder simplifies the AI creation experience by giving people with any level of technical skill the ability to add AI capabilities to their apps and flows without writing code. AI Builder also provides prebuilt AI models, where you don't need to gather data to build and train the model. You can start to use the intelligence right away.

### What you can do with AI Builder

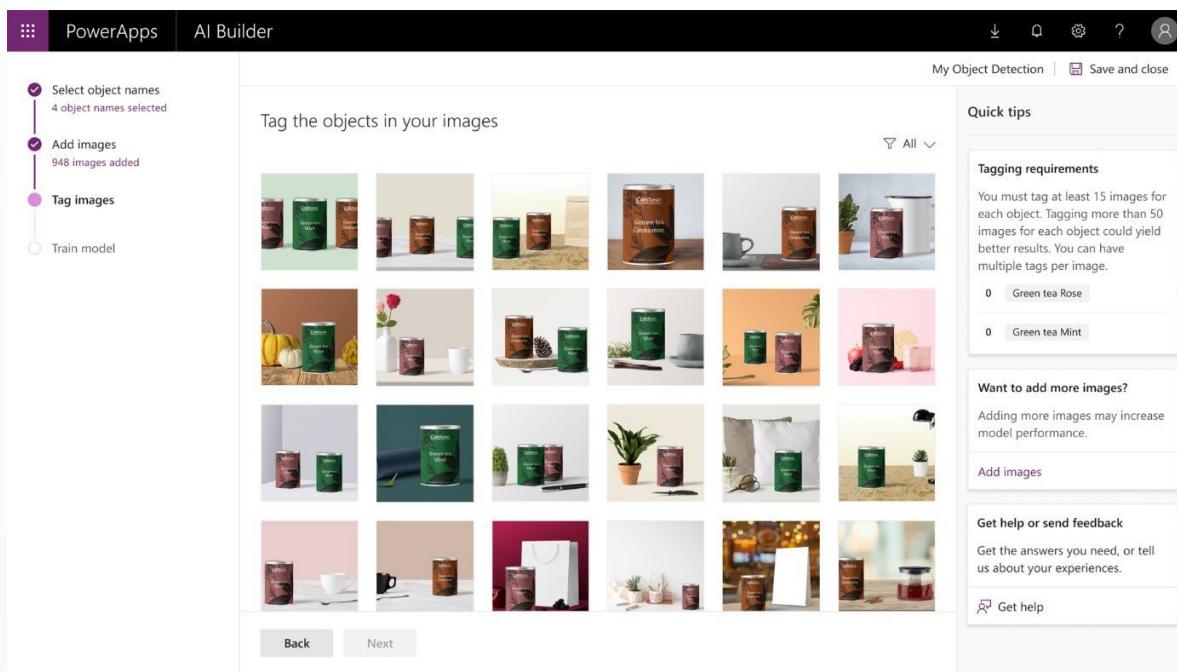
You can use AI Builder to create custom AI models that suit the needs of your business, or you can choose from a selection of prebuilt models.

You can then use the AI from these models in your apps and workflows.

For example, with AI Builder you can:

- Analyze text for classification, key phrases, language, and sentiment.

- Predict whether something will happen.
- Process business card information.
- Process text from images.
- Read and save information from standard documents.
- Recognize and count items in images.



A common use of AI Builder is to automate processes. The following examples can help you think about how you can use AI in your business:

- **Invoice processing** - Companies often receive invoices in large quantities and from a variety of sources, such as mail, fax, email, or in person. Processing these documents and manually entering them into your database can take considerable time. By using AI to extract the text, key/value pairs, and tables from your documents, you can create workflows to automatically pipe the information into your database. You can even create an app to review the information.
- **Text analysis** - Businesses today are collecting more data than ever before. Whether it's from survey responses, product reviews, user feedback, or support emails, generating the proper insights is important to understanding your business and your customers. By using AI to analyze text, you can identify customer feedback that contains negative sentiment or certain key phrases. Use AI to identify problems and then trigger the necessary workflows to take action.

## Where you can use AI Builder

You can access AI Builder within Power Apps and Power Automate.

The screenshot shows the Power Apps AI Builder interface. On the left, there's a navigation menu with items like Home, Learn, Apps, Create, Data, Flows, AI Builder (which is currently selected), Build, Models, and Solutions. The main content area has a header "Enhance your business with AI" and a sub-header "Refine a model for your business needs". It displays four cards: Form Processing (preview), Object Detection (preview), Prediction (preview), and Text Classification (preview). Below this, another section titled "Get straight to productivity" shows four more cards: Business Card Reader (preview), Key Phrase Extraction (preview), Language Detection (preview), and Sentiment Analysis (preview). Each card includes a small icon, a title, and a brief description.

Two available areas under AI Builder in the left side menu are:

- **Build** - Where you create and get started by using AI models.
- **Models** - Where your created and shared models reside.

## Next steps

Now that you've learned about AI Builder, your next step is to gain in-depth knowledge about the available AI capabilities.

## Choose an AI capability

AI Builder offers several AI capabilities.

AI capabilities are brought to your apps and flows by models. A model can be built and customized by you, or it can be a prebuilt model that is ready to be used right away. Before learning about the model creation process, you should be familiar with the various model types that are available.

## Choose a model

AI Builder comes with a wide variety of models to enhance your apps and business processes.

1. Sign in to Power Apps or Power Automate and select **AI Builder Build** on the left menu. Each tile represents a different AI capability that you can bring to your business.

The screenshot shows the Power Apps AI Builder interface. On the left is a navigation sidebar with options like Home, Learn, Apps, Create, Data, Flows, AI Builder, Build, Models, and Solutions. The AI Builder section is currently selected. The main area has a header "Enhance your business with AI" and sub-sections "Refine a model for your business needs" and "Get straight to productivity". Under "Refine a model for your business needs", there are four tiles: "Form Processing (preview)", "Object Detection (preview)", "Prediction (preview)", and "Text Classification (preview)". Under "Get straight to productivity", there are four tiles: "Business Card Reader (preview)", "Key Phrase Extraction (preview)", "Language Detection (preview)", and "Sentiment Analysis (preview)". Each tile includes a small icon, a title, and a brief description.

2. Select any tile.

## Learn about the model

Each AI Builder model has an introductory experience where you can learn more about the model.

The screenshot shows the "Form Processing (preview)" model details. A modal window is open over the main interface. The modal has tabs "Get started", "Examples", and "Best practices", with "Get started" selected. It displays a diagram of a document structure with fields like "Contoso, Ltd.", "1/28/2020", "(\$5.99)", and "\$993.20". Below the diagram, it says "Automatically process documents" and "Save time on routine documents like invoices or tax forms. Use AI to review, extract, organize, and save their data automatically.". To the right of the diagram, there's a form field "Name this AI model \*". Below it, "What you'll need" is listed as "5+ documents with the same layout". At the bottom of the modal are "Create" and "Cancel" buttons.

For those models that require customization, AI Builder provides a summary of the steps that you need to take and what you need to get started. The **Examples** tab also includes examples of how you can use the model.

For prebuilt models, AI Builder includes best practices on how to use them and buttons that you can select to get started using them right away.

## Next steps

Now that you have learned about AI capabilities, your next step is to learn how models are created and managed.

## Create your first model

You can create a model in AI Builder by following a few simple steps.

### Select a model type

Sign in to Power Apps or Power Automate and select **AI Builder** **Build** on the left menu.

The two classes of model are:

- Models that require customization before they can be used
- Models that are prebuilt and can be used right away

Prebuilt models have already been created for you and can be used right away in your apps and flows. No prior work is needed to create the model.

The following sections explore what you can do with the models that you customize.

Select the tile for one of the following model types:

- Form processing
- Object detection
- Prediction
- Text classification

Make sure that you have everything you need to create your model, as listed on the **Get started** tab. You can also use **sample**

**data<sup>1</sup>** to create any of these models.

1. Enter a name for your model in the **Name this AI model** column.
2. Select **Create**.

## Follow the guided experience

To help create your model, AI Builder offers a guided experience that will walk through each step of the experience.

<sup>1</sup> <https://docs.microsoft.com/ai-builder/samples>

Every model type has its own set of steps to create a model. You can find these steps outlined in the left column to help guide you through the model creation process. As you progress from step to step, the left column will be updated to reflect your progress.

At any time during the creation process, you can save your work and come back later. Select **Save and close** in the upper-right corner, next to the name of your model. Progress is also saved automatically when you go between steps.

Along the right column is a list of **Quick tips** that can help you understand some of the concepts and actions that you need to take in the current step. If you need more help, select the **Get help** link on the **Get help or send feedback** tip card, where you can report an issue, send feedback, and find documentation.

Follow the instructions on each page to advance to the next step.

The final step is a summary of the customizations that you made in the previous steps. Review this information to be sure that every detail is configured as intended, and then begin training your model. Training might take a while depending on the size of your data.

After the training has been completed, your model is ready to be tested.

## Next steps

You have now learned how to create your first model. Next, you'll learn how to use your model in Power Apps and Power Automate.

## Ways to use your models

After creating your AI Builder model, you can use it in Power Apps and Power Automate.

## View your model details

After your model has completed training, you can view important details about your newly trained model on a details page for that model. The information might vary depending on the model type.

The screenshot shows the 'My Form Processing model' details page. On the left, the navigation menu includes Home, Learn, Apps, Create, Data, Flows, AI Builder, Build, Models (which is selected), and Solutions. The main content area displays a 'Training document' preview showing an invoice form, followed by a list of 'Selected fields' and 'How your model is used'.

On the model details page, you can see the customizations that you made to train your model. In some cases, it shows additional insights on the training performance. Some model types give you the opportunity to quick test your model to see it live in action.

You can access this page at any time from the left menu by selecting **AI Builder Models** and then searching for your model name.

## Publish your model

Your model can't be used until it is published. If you are satisfied with your model, select **Publish** to make it available.

Three main ways that you can use your model are:

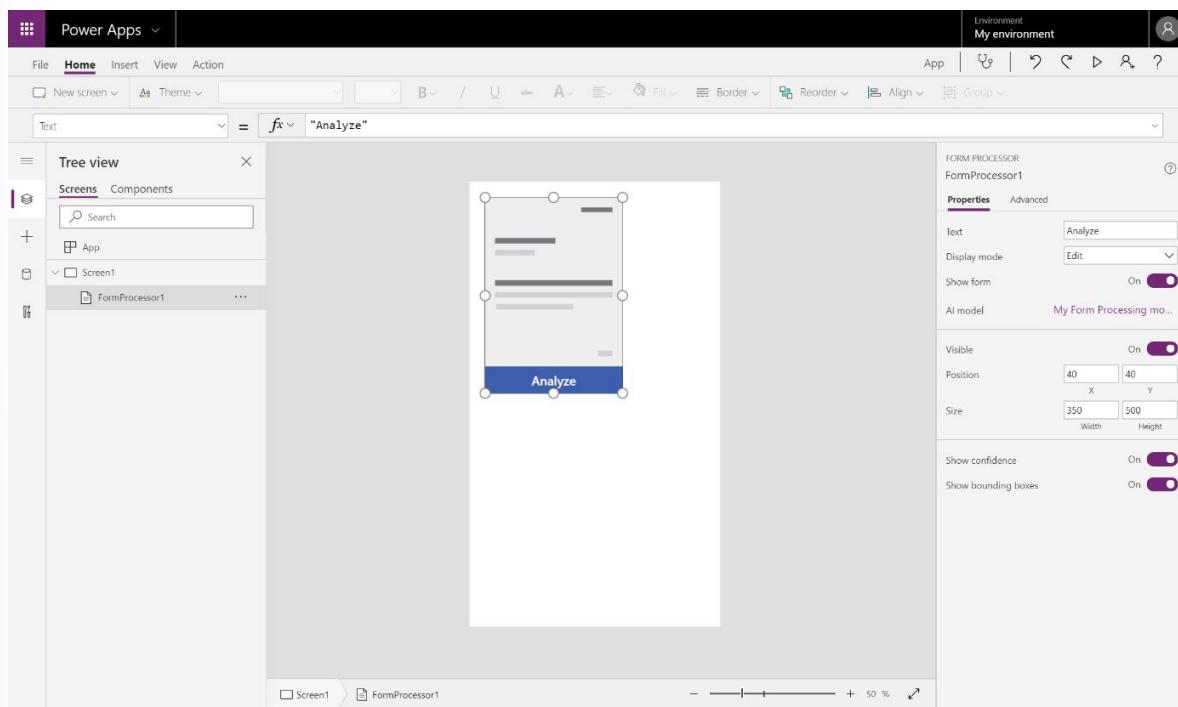
- As a component in an app
- As an action in a flow
- As new data in your database

When your model is published, select **Use model** to see a list of the available actions that you can take to use your model.

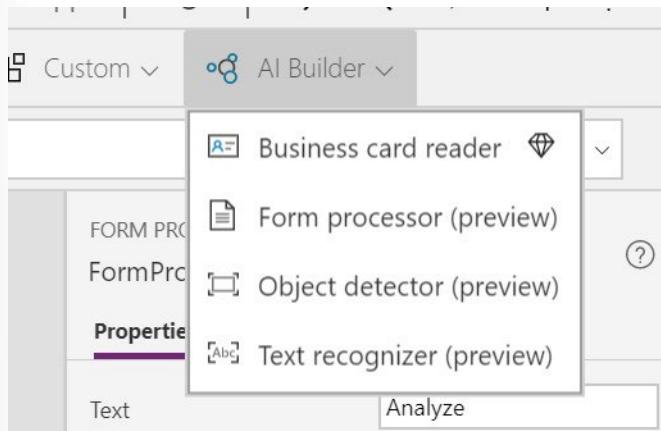
## Use your model in an app

In the **Use your model** pane, select **Create new app**, which appears if your model type supports it.

This selection redirects you to the canvas app creation experience, with the AI Builder component already added to your canvas and your model automatically linked to the component.



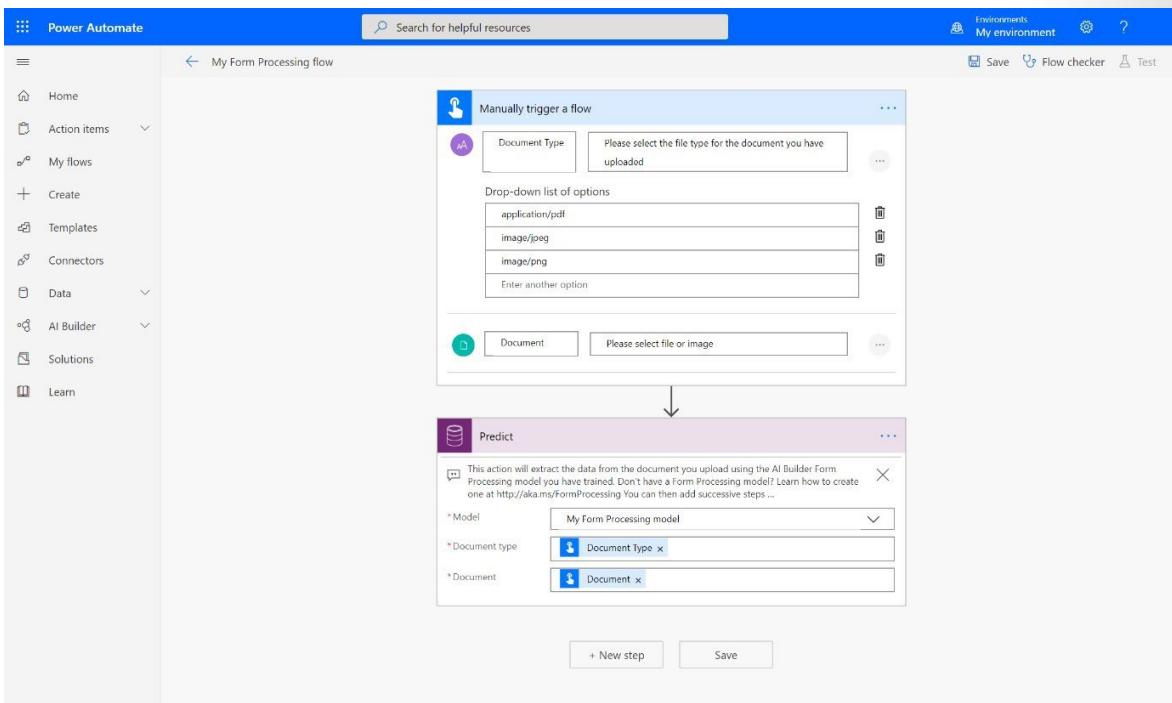
You can add AI Builder components to your existing apps at any time by selecting the **Insert** tab and then selecting the component from the **AI Builder** menu.



## Use your model in a flow

In the **Use your model** pane, select **Create new flow**, which appears if your model type supports it.

This selection redirects you to the flow template page in Power Automate. Confirm the connections and then select **Continue**.



In the flow creation experience, you'll find the AI Builder action already added to your flow and your model automatically linked to the action.

You can add the AI Builder action to **solution-aware flows<sup>2</sup>**

by searching for the **Predict** action under **Dataverse (current environment)** and then selecting your model name from the **Model** drop-down menu.

## Use your model in your database

Some model types write the intelligence back to your database, so you can use it in your data views in Power Apps or Power BI.

After you have published your model, some model types automatically begin scheduling the model to write data back to your database by default. For others, you can customize the scheduling. In the **Use your model** pane, select **Set run schedule** to define the frequency. This option appears when the model supports it.

You have now learned the various ways that you can use your AI Builder models in Power Apps and Power Automate.

## Summary

Congratulations on creating your first model with AI Builder.

In this module, you discovered how to bring AI to your business by using AI Builder.

Specifically, you learned that:

- A wide variety of AI capabilities are at your disposal.

<sup>2</sup> <https://docs.microsoft.com/power-automate/create-flow-solution>

- Some model types can be customized to your needs, whereas other model types can be used right away.
- You can use your model in several different ways: in your apps, in your flows, and in your database.

# Create a chatbot with Power Virtual Agents for Teams

## Introduction

Employees who have quick access to information make better decisions, reduce errors, and save money. Chatbots create a personalized experience that simplifies how employees get information. Microsoft Power Virtual Agents and Microsoft Dataverse for Teams create intelligent chatbots to quickly respond to employee needs without building high-code solutions.

By the end of this course, you will be able to:

- Create a chatbot.
- Use the chatbot to call Microsoft Power Automate actions to retrieve data from Dataverse for Teams.
- Publish the chatbot to your team or the whole company.

This module goes through the process of creating, editing, and then publishing a chatbot that provides contact information for an event planning team.

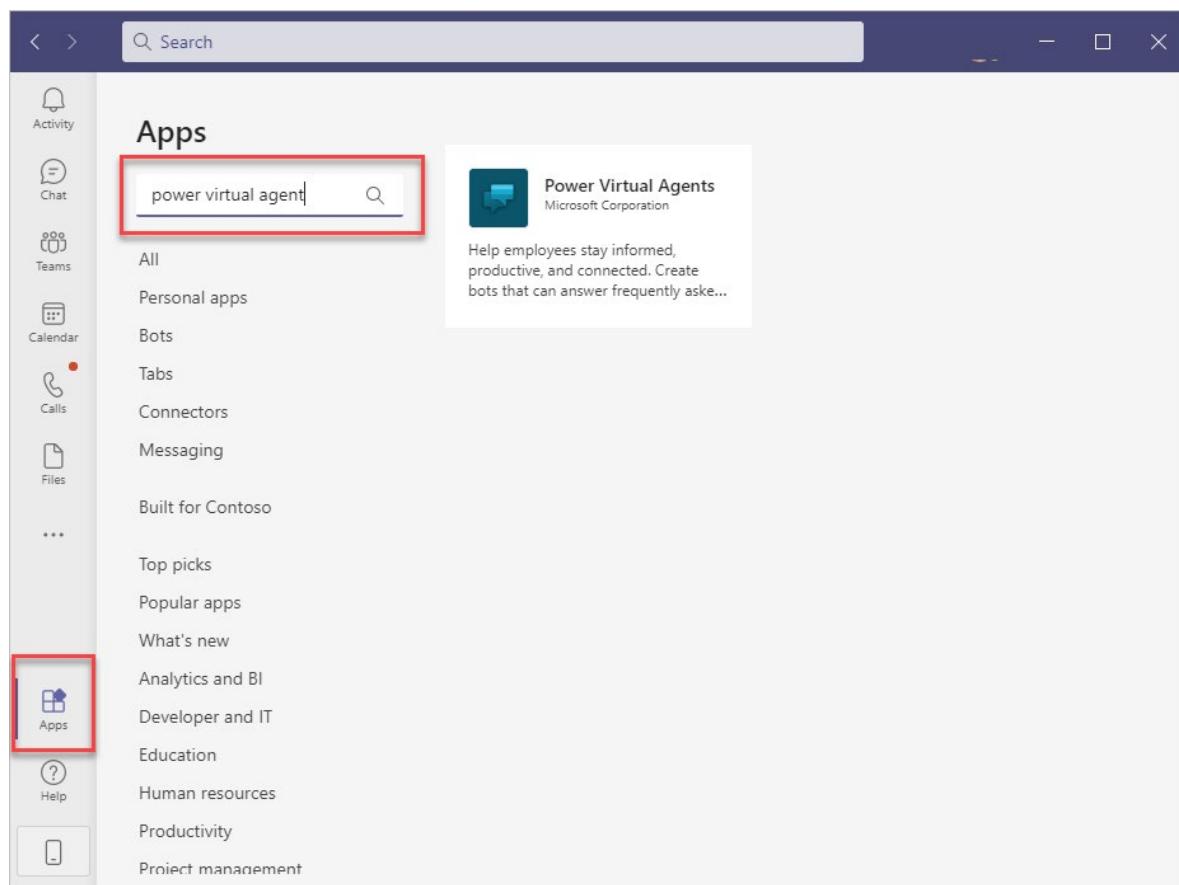
## Create your first chatbot

To create chatbots (also referred to as *bots*), first install the Power Virtual Agents app in Microsoft Teams. Power Virtual Agents allows you to:

- Create and edit chatbots.
- Test the chatbot to see how well it works.
- Publish the chatbot to your team or organization.
- Review the bot's performance over time, note the questions that are asked, and assess how well it's answering questions.

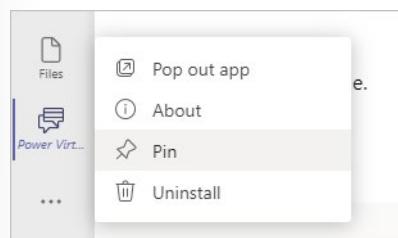
To install Power Virtual Agents for Teams:

1. Open Teams and then select **Apps** from the left navigation bar.
2. Enter **Power Virtual Agents** in the search box and then select it from the search results.



3. Select **Add** to install the Power Virtual Agents app in Teams.

**Note:** Pin the **Power Virtual Agents** app in Teams for easier access. Select the app by selecting the ellipsis (...) and then selecting **Power Virtual Agents**. Right-click the **Power Virtual Agents** icon and select **Pin**.



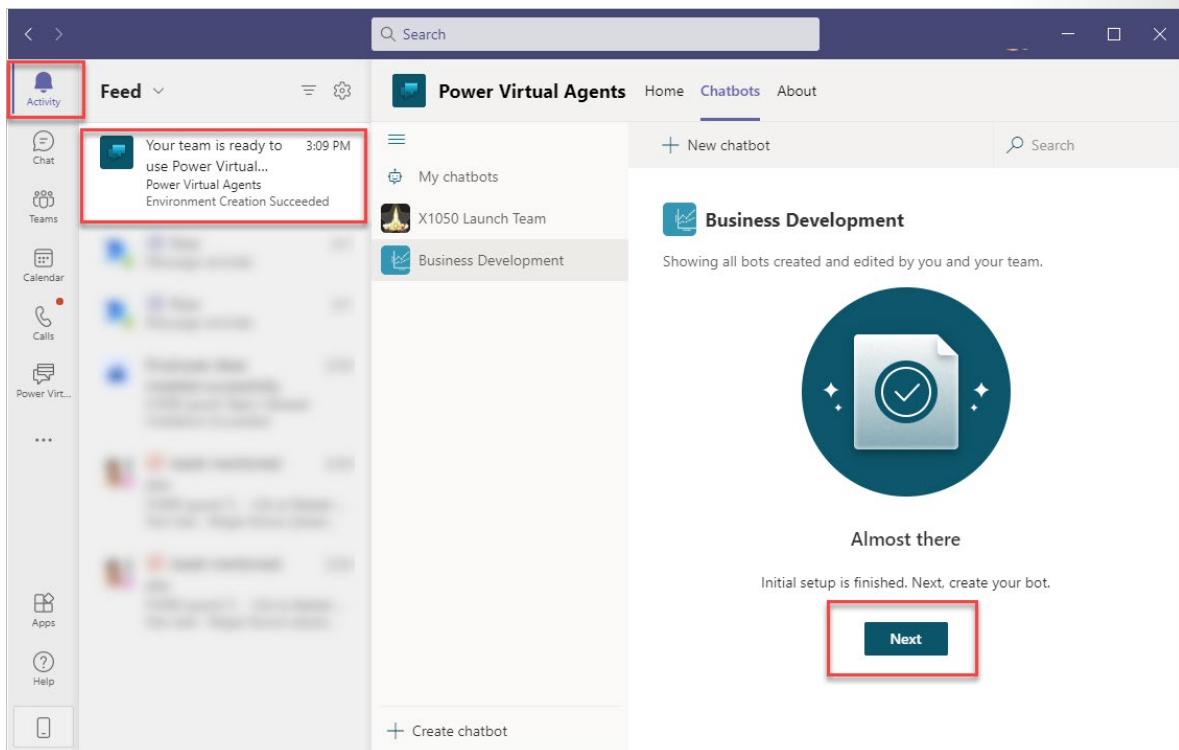
## Create a new chatbot

To create a new chatbot, follow these steps:

1. Open the Power Virtual Agents for Teams app.
2. Select **Start now**.
3. Select the team to own and manage your chatbot from the drop-down list and then select **Continue**.

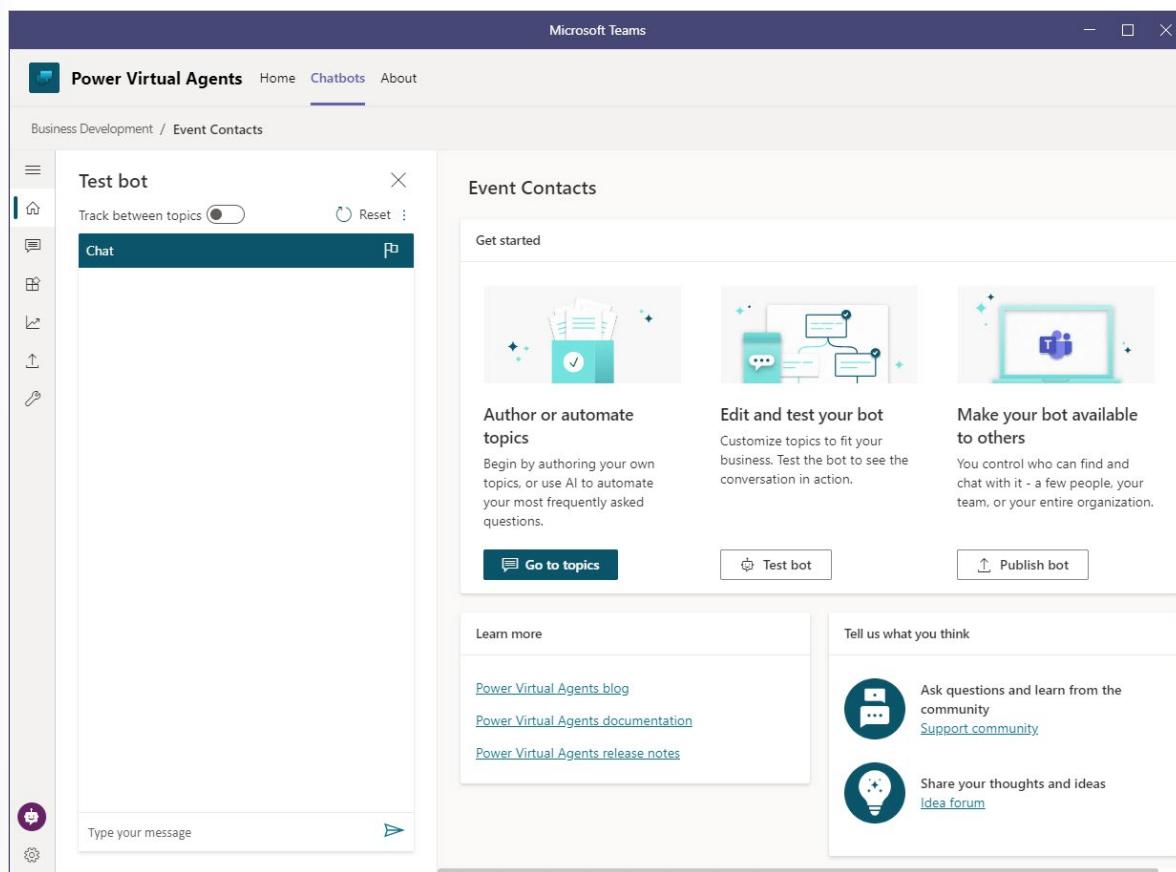
When you are creating a new chatbot, team owners and the bot creator have read/write access by default. Team members only have read access to the chatbot.

If this is your first time creating a chatbot or Power Apps application in the team, you will receive a notification that the process will be slightly longer than usual while the system sets up. Select **Close** to continue and then wait a few minutes for Teams to notify you that the chatbot is ready. After setup has completed, open the notification under **Activity** and then select **Next**.



4. Provide a **Name** and select the **Language** for the chatbot. For this example, use **Event Contacts** for the name. After you have set the chatbot's language, it can't be changed.
5. Select **Create** to complete creating a new chatbot.

From the Power Virtual Agents editor, you can manage topics, edit and test your chatbot, and then publish it to the rest of the team or the entire organization. Topics are how chatbot conversations transpire and they are covered in the next section.



## Create and modify topics

In Power Virtual Agents, chatbot conversations are defined by *topics*. With topics, you can define and control the way that the conversation evolves.

Two types of topics are:

- **Trigger phrases** - The chatbot needs to detect when the user asks a question that it knows how to respond to. The trigger phrases are the phrases, keywords, and questions that the user is likely to enter. We recommend that you have 5 to 10 trigger phrases of common ways that your users would request help on the topic.
- **Conversation nodes** - When the chatbot has a question, it needs to know how to handle the request. Conversation nodes define how the chatbot responds and if actions are required.

To view and edit topics in your chatbot, select **Topics** from the left column in the Power Virtual Agents editor. Many topics are automatically created for a chatbot:

- The first four topics are sample **user topics** that demonstrate various ways of using topics to create conversations.
- The remaining topics are **system topics** that are likely needed during a chatbot conversation.

## Change the greeting system topic

System topics improve the usability of your chatbot with prebuilt topics to handle greeting, escalating, starting over, and saying goodbye. Editing these topics allows you to further personalize the chatbot for your users and organization.

The **Greeting** system topic provides a welcome message for the user. It's important to clearly define what the bot is meant to do, which will set expectations with the user.

1. Select the **Greeting** system topic and then select **Go to authoring canvas**.

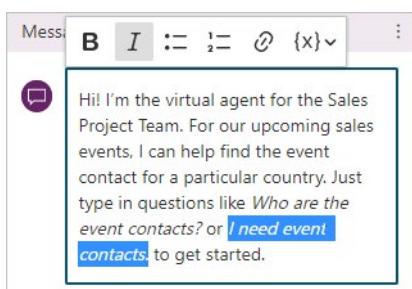
The authoring canvas is where you design the topic's conversation path by using *conversation nodes*. Conversation nodes determine how a chatbot responds and what actions it might have to take. The five different types of conversation nodes that you can add are described in the following table.

| Conversation node | Description                                                      |
|-------------------|------------------------------------------------------------------|
| Ask a question    | Have the chatbot ask a question and get a response from the user |
| Add a condition   | Add branching logic                                              |
| Call an action    | Call Power Automate flows                                        |
| Show a message    | Have the chatbot respond to the user                             |
| End with survey   | When the conversation ends, a survey appears                     |

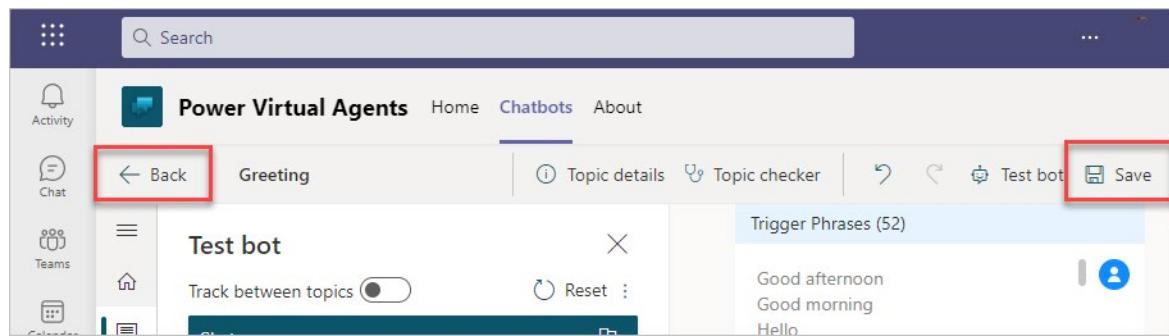
All chatbots start with the trigger phrases and a message conversation node.

2. The text in the first message node doesn't match the purpose of your bot. Change the text to the following example. Use the text editing controls in the message node to have the questions appear in italic font.

Hi! I'm the virtual agent for the Sales Project Team. For our upcoming sales events, I can help find the event contact for a particular country. Just type in questions like *Who are the event contacts?* or *I need event contacts.* to get started.



3. Save the settings by selecting **Save**.
4. Exit the topic authoring canvas by selecting **Back**.



## Create a topic

For the **Event Contacts** chatbot, you want the chatbot to respond to the question, "Who are the event contacts?" From the **Topics** page in Power Virtual Agents in Teams:

1. Select **+ New topic**.
2. Enter **Contacts** for the **Name**.
3. Enter the following **Trigger phrases**. Select **Add** between each one.

who are the event contacts  
event contacts  
who to contact for the event  
event contact information  
event info

A screenshot of the 'Setup' tab in the Power Virtual Agents Topics page. The 'Name' field is set to 'Contacts'. The 'Trigger phrases' section contains five entries: 'event info', 'event contact information', 'who to contact for event', 'event contacts', and 'who are the event contacts'. An 'Add' button is located next to the input field. Other tabs like 'Analytics' and 'Go to authoring canvas' are visible.

4. Select **Save topic** in the upper-right corner to add the topic to the list.
5. Select **Go to authoring canvas**. In the **Message** node, enter the following information. Use the editing options in the text box to apply bold formatting to the country names.

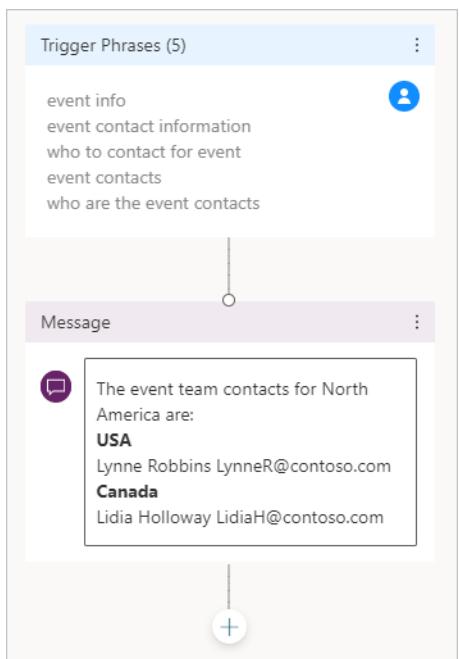
The event team contacts for North America are:

**USA**

Lynne Robbins LynneR@contoso.com

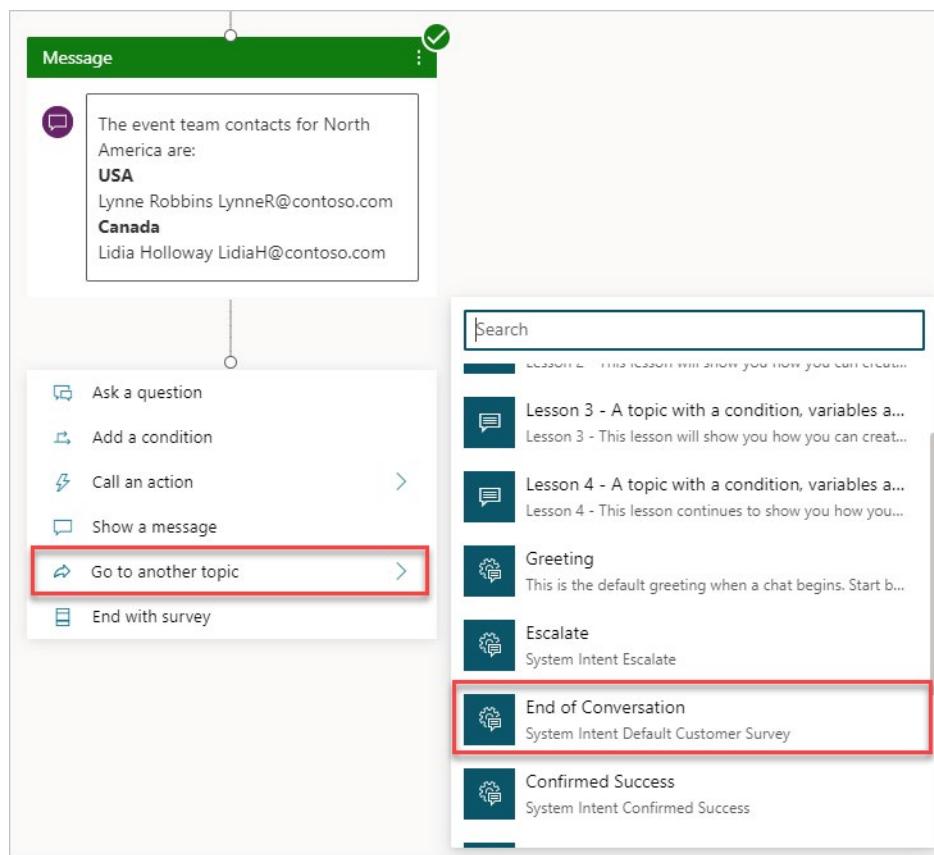
**Canada**

Lidia Holloway Lidia@contoso.com

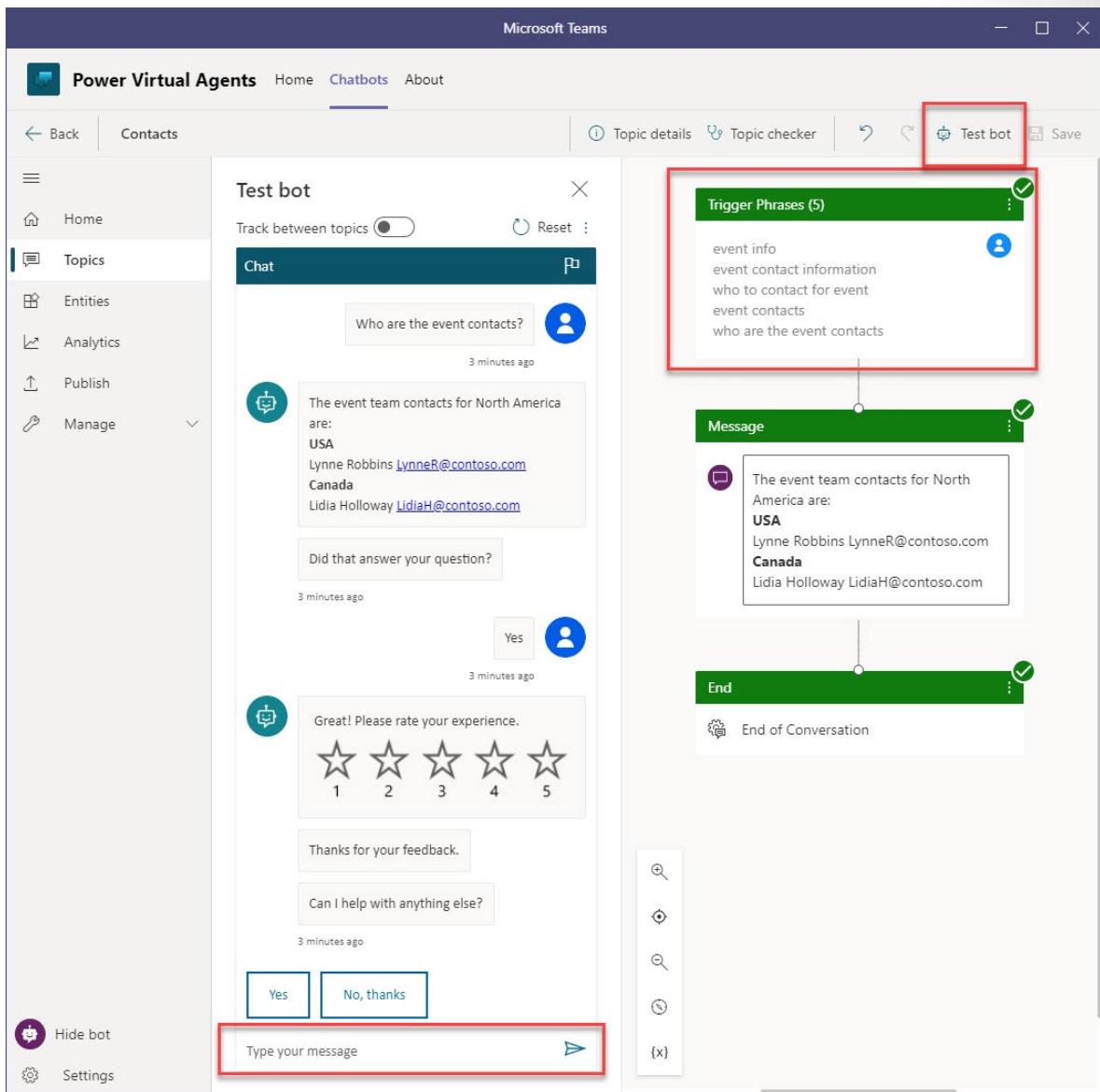


6. Select the plus + symbol under the **Message** node to add another conversation node.
7. Select **Go to another topic > End of Conversation**.

The **End of Conversation** option is a system topic on how to end the conversation. This system topic asks the user about the accuracy of the results and to rate their experience. All system topics can be edited to customize the experience in the **Topics** screen.



8. Select **Save**.
9. Select **Test bot** and a new pane will appear for testing the chatbot. Enter questions that contain one of the trigger phrases to observe how the bot responds.



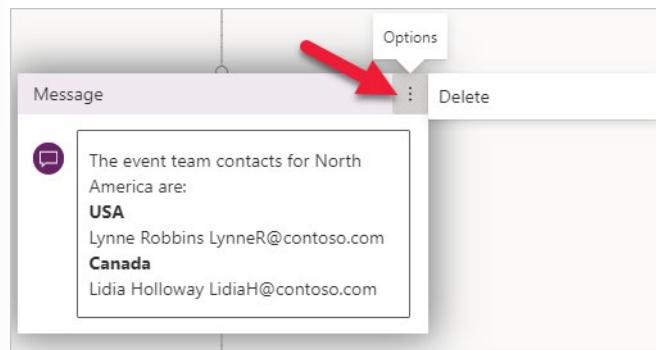
Congratulations, you have created your first chatbot by using Power Virtual Agents for Teams. The next unit expands the chatbot to use inputs, variables, and conditions.

## Add inputs, variables, and conditions

For your **Event Contacts** chatbot, you need to expand the number of countries that have contacts. Instead of listing all countries in the message node, you want to ask the user which country they need contact information for. To accomplish this task, you will use inputs, variables, and conditions.

- **Inputs** - The user's response in the **Ask a question** node.
- **Variables** - Store the **Inputs** to use in later conversation nodes.
- **Conditions** - Define the branching logic based on variables.

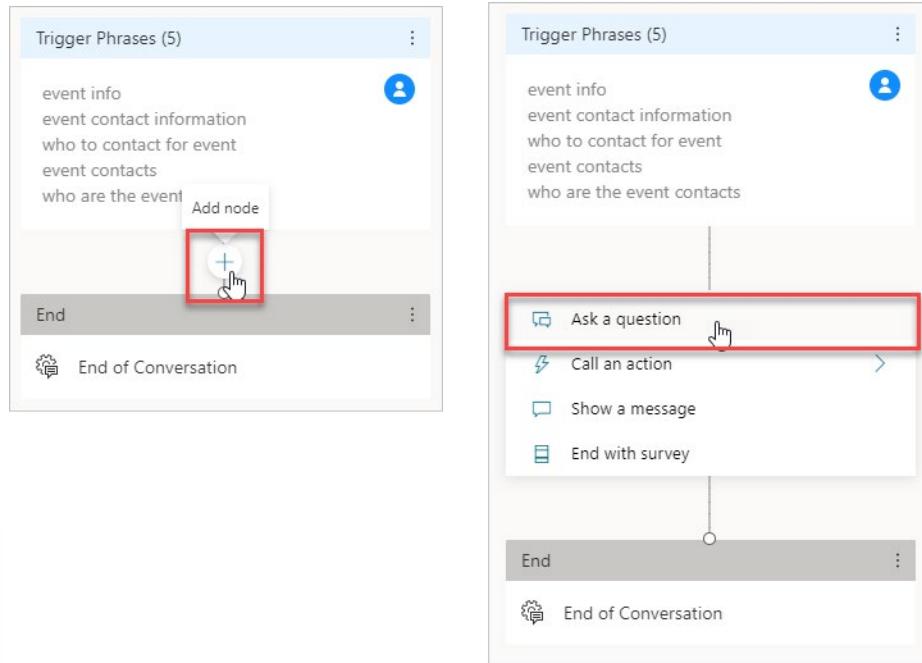
For your chatbot, you need to delete the current message node because it will be replaced. Select the options ellipsis (...) for the message node and then select **Delete**.



## Add inputs and variables with the question node

To add inputs and variables with the question node, follow these steps:

1. Use the mouse to hover over the line that connects the **Trigger Phrases** to the **End** node. Select the plus + sign that appears and then select **Ask a question**.

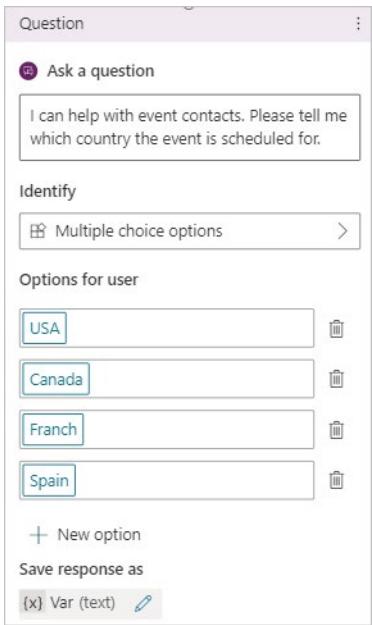


2. Fill out the question node with the following information:

**Ask a question** - Enter the phrase, "I can help with event contacts. Please tell me which country the event is scheduled for."

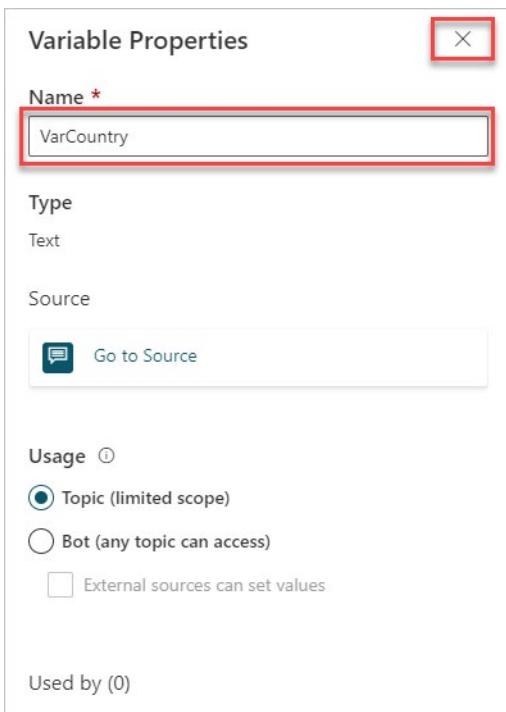
**Identify** - Select **Multiple choice options**.

**Options for user** - Enter **USA**, **Canada**, **France**, and **Spain** by selecting **+ New option** between each word.



The **Identify** field is where you indicate what type of response that the chatbot is listening for. Other responses that you can select include a number or a string. You can even have the chatbot detect entities such as state names, phone numbers, or street addresses. To learn more about the different available options, see the links in the Summary unit at the end of this module.

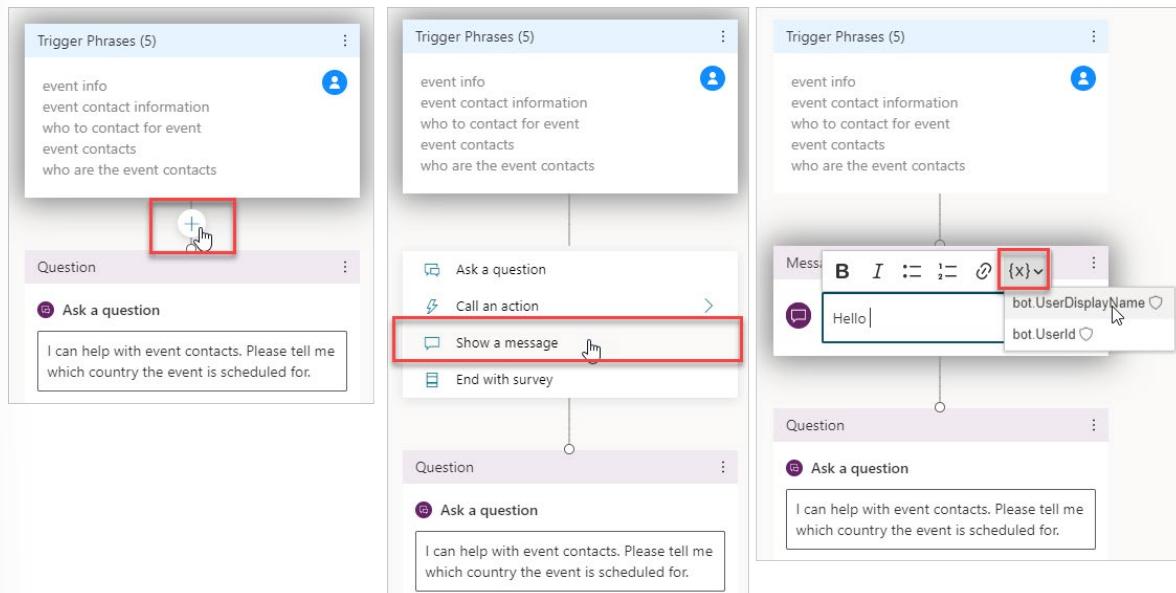
- To set the **variable** name, select the text under **Save response as** and replace **Var** with **VarCountry**. Select the **X** to close the **Variable Properties** screen. The variable name stores the user's selection and will be used when the user configures the branching logic under the condition node.



Two variables are available by default: **bot.UserDisplayName**, which provides the user's name, and **bot.UserId**, which provides the user's sign-in name. By using **bot.UserDisplayName**, the chatbot can call the

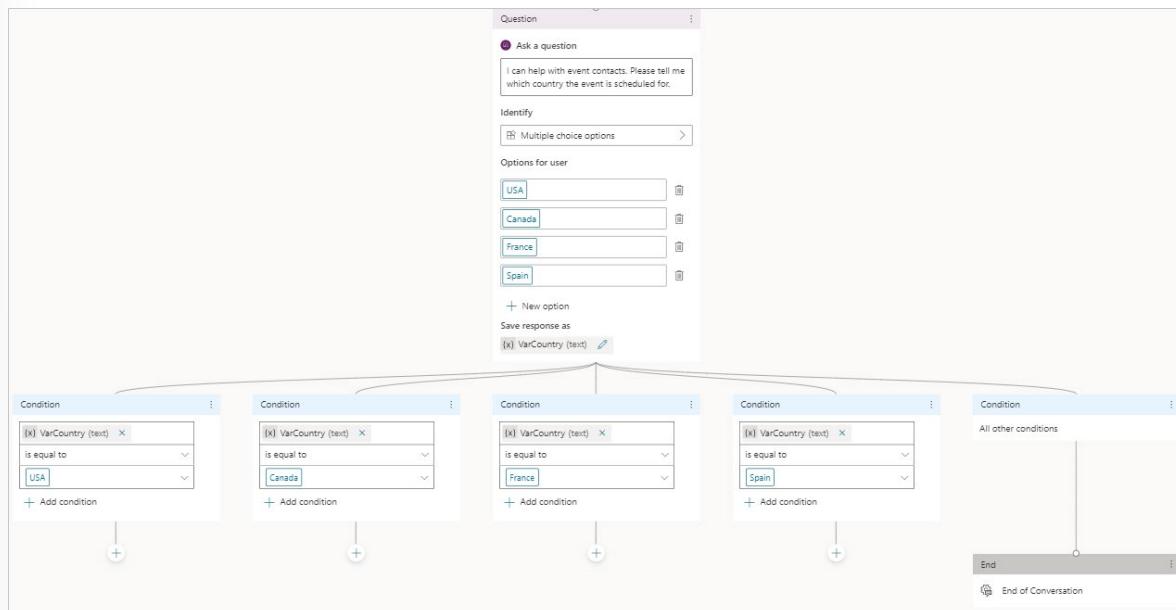
user by name, which personalizes the experience. You can use **bot.UserID** to create a condition node to take actions based on which user whom the bot is interacting with.

To have the **Event Contacts** chatbot greet the user by name, add a new message node at the chatbot's beginning. When entering information in the message node, select the **{X} insert variable** drop-down menu and select **bot.UserDisplayName**.



## Condition node

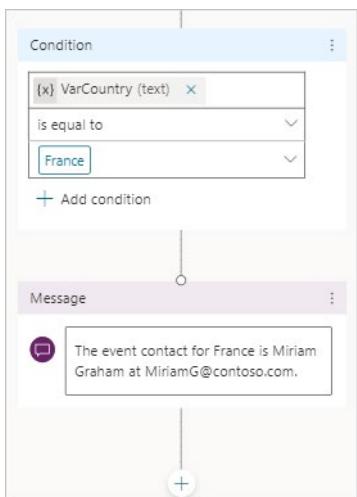
For each multiple-choice option in the question node, Power Virtual Agents creates a new condition node. You need to configure each condition node to provide the next appropriate response in the conversation.



For your **Event Contacts** chatbot, you need to respond with the event contact for the country. For example, if the user selects **USA**, then the chatbot should respond with Lynne Robbins' contact information.

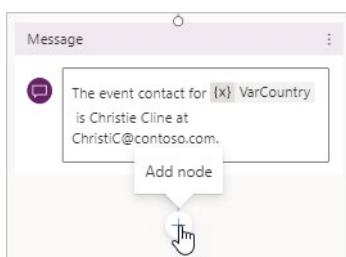
| Country | Name           | Email                 |
|---------|----------------|-----------------------|
| USA     | Lynne Robbins  | LynneR@contoso.com    |
| Canada  | Lidia Holloway | LidiaH@contoso.com    |
| France  | Miriam Graham  | MiriamG@contoso.com   |
| Spain   | Christie Cline | ChristieC@contoso.com |

For each condition node, add a new message node after the condition that provides the correct contact and email address. For example, "The event coordinator for France is Miriam Graham at `MiriamG@contoso.com`."

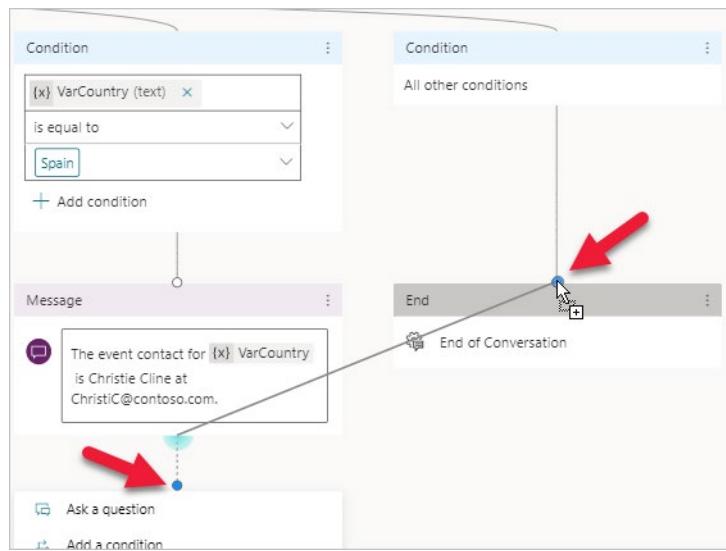


For all four message nodes, you want to use the same **End of Conversation** system topic. Your chatbot already has this topic defined under the **All other conditions** node. To link other nodes to this one, perform the following steps:

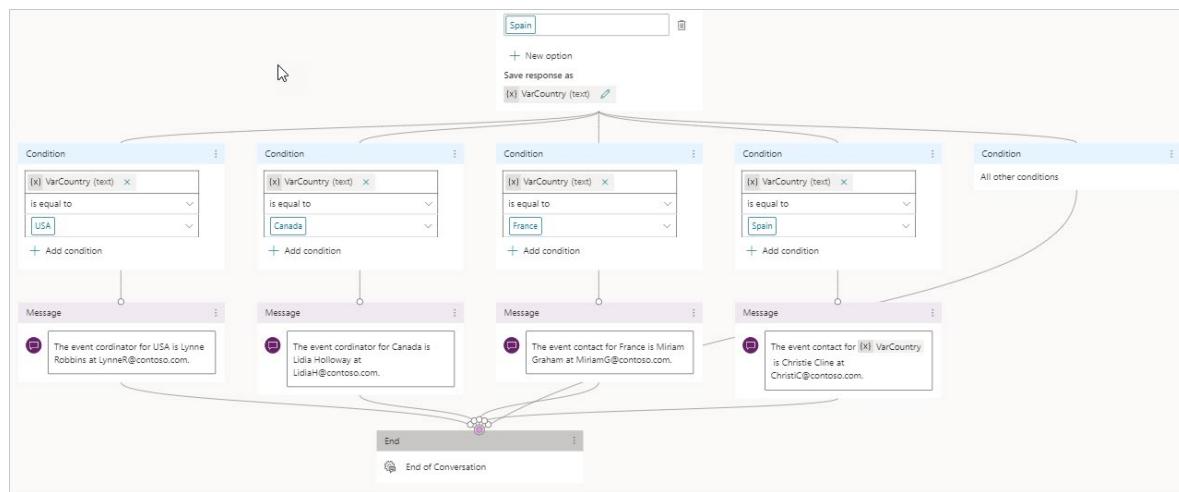
1. After one of the message nodes, select the **Add node** plus + symbol.



2. Press and hold the left mouse button on the point where the line connects to the new node and drag it over to the same spot on the **End** node.



3. The editor adjusts the nodes to show that both are connected. Repeat the same steps for the other three message nodes. Your editor should now look like the following image.



To test the chatbot, select **Save**. Select the **Test bot** option if the chatbot testing pane isn't visible.

## Call an action to pull in Dataverse in Teams data

With the **Call an action** node, chatbots can interact with Power Automate flows. You can use these flows to trigger automated processes or connect with Dataverse for Teams data, SharePoint lists, or other data sources.

For example, the **Event Contacts** chatbot could provide the Sales Project team members from a Dataverse for Teams table. Instead of editing the chatbot when changes are made to this team, you can update the table.

## Step 1 - Create a Dataverse for Teams table

To create a Dataverse for Teams table, follow these steps:

1. Create a new Dataverse for Teams table in the same team as the chatbot, named **Sales Project Team**.  
The columns and data types to use are as follows:

| Column        | Data type |
|---------------|-----------|
| Name          | Text      |
| Email Address | Email     |

2. Add new rows for each of the four contacts.

| Name            | Email Address      |
|-----------------|--------------------|
| Paul Cannon     | paul@contoso.com   |
| Toby Nixon      | toby@contoso.com   |
| Yukari Kemmotsu | yukari@contoso.com |
| Zac Woodall     | zac@contoso.com    |
| Enter text      | Enter email        |

3. Close the table when you are finished.

## Step 2 - Add an action

To add an action, follow these steps:

1. Open the **Event Contact** chatbot in Power Virtual Agents for editing.
2. Create a new topic.

**Name** - Sales Project Team

**Trigger phrases** - sales project team, sales project info, sales team info, sales project members, sales project info

The screenshot shows the 'Setup' tab of a Microsoft Bot Framework configuration page. On the left, there's a form with fields for 'Name \*' (containing 'Sales Project Team'), 'Friendly name (Optional)', and 'Description (Optional)'. In the center, a section titled 'Trigger phrases (5)' displays five entries: 'sales project info', 'sales project members', 'sales team info', 'sales project info', and 'sales project team'. A red box highlights the 'Name \*' field and the list of trigger phrases. On the right, a button 'Go to authoring canvas' is visible.

3. Select **Save topic** and then select **Go to authoring canvas**.
4. In the **Message** node, enter **Let me look up the Sales Project Team!**
5. Select **Save** to save the bot at this point. When the Power Automate flow is created, you will switch to a Power Automate editor. Changes that are not saved to the chatbot could be lost.
6. Add a new node after the message node and then select **Call an action > Create a flow**.

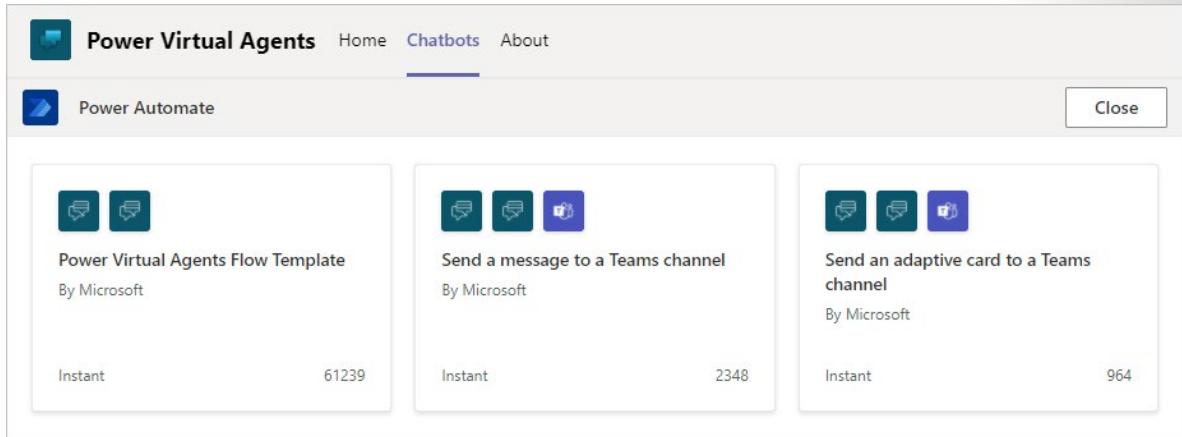
The screenshot shows the Power Virtual Agents authoring canvas. On the left, a list of actions includes 'Ask a question', 'Add a condition', 'Call an action' (which is highlighted with a red box), 'Show a message', 'Go to another topic', and 'End with survey'. On the right, a search bar and a list of available actions are shown, with 'Authenticate' and 'Create a flow' listed. 'Create a flow' is also highlighted with a red box. Below the list, it says 'No actions found.'

Three templates are available for you to choose from. Each template allows you to pass information to the flow when it is triggered. After the flow has finished, information can be passed back to the chatbot.

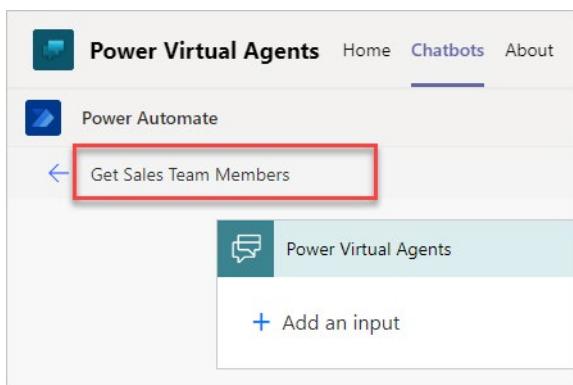
**Power Virtual Agents Flow Template** - This template only has the steps to receive and send information to the chatbot.

**Send a message to a Teams channel** - Trigger a Teams message from the chatbot.

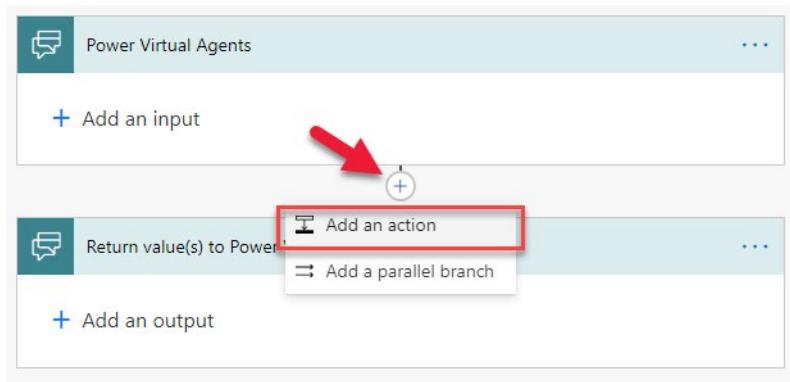
**Send an adaptive card to a Teams channel** - Trigger an adaptive card to provide information into a Teams channel.



7. Select **Power Virtual Agents Flow Template**.
8. The title of the new flow is **Power Virtual Agents Flow Template**. Select the title of the flow to rename it to **Get Sales Team Members**.

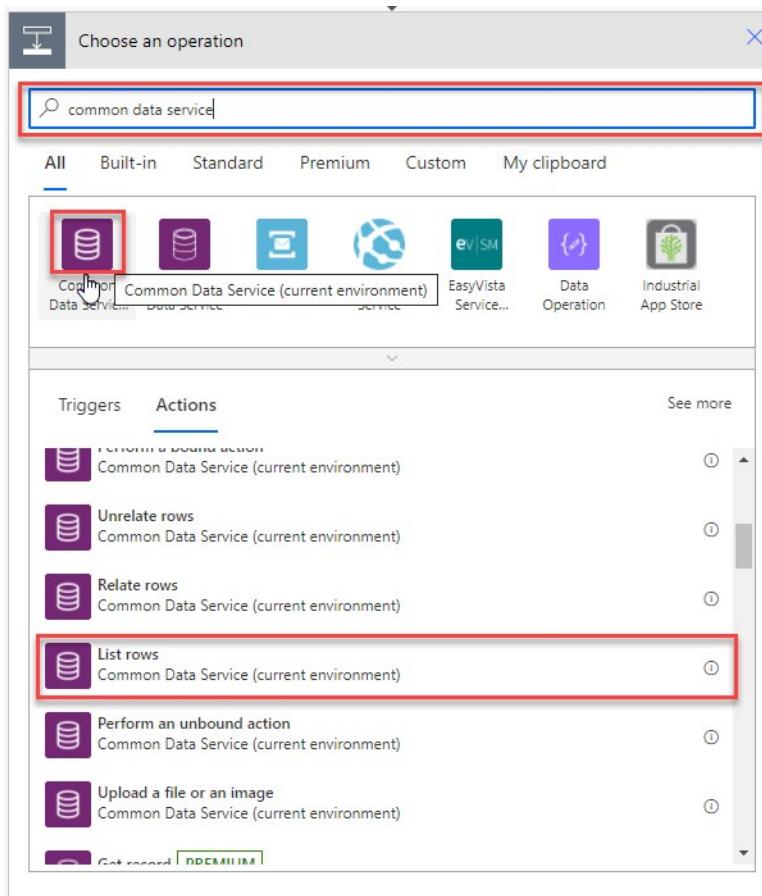


9. Create a variable to hold the information that you want to provide back to the chatbot. Select the **Insert new step** icon and select **Add an action**.



10. In the search box, enter **variable** and then, under **Actions**, select **Initialize variable**.
11. Enter **ContactInfo** for the **Name** and then select **String** for the type.

12. Now we need a step to access the data from Dataverse from Teams. Under the **Initialize variable** step, select the **Insert new step** icon and select **Add an action**. Search for the **Common Data Service (Current Environment)** connector then select **List Rows** as the trigger.



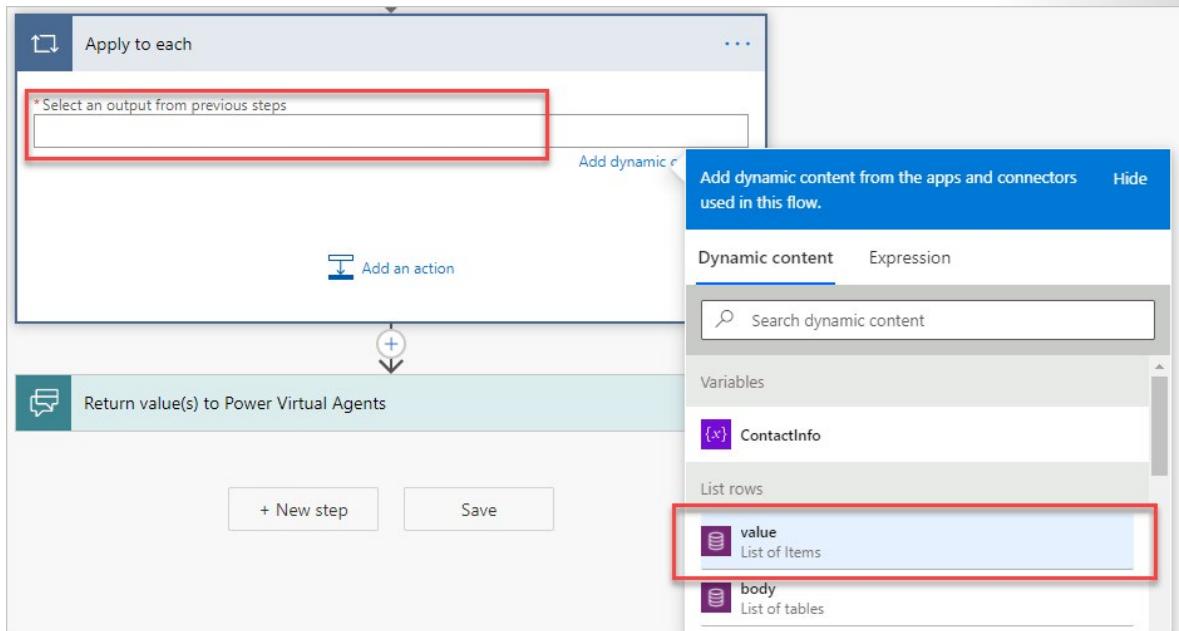
13. From the drop-down list, select the **Sales Project Team** table.

The **List rows** action pulls all table rows into an array. An array is a type of variable that can hold more than one value at a time. To work with data in the array, you need to go through each row and add the data to your **ContactInfo** variable.

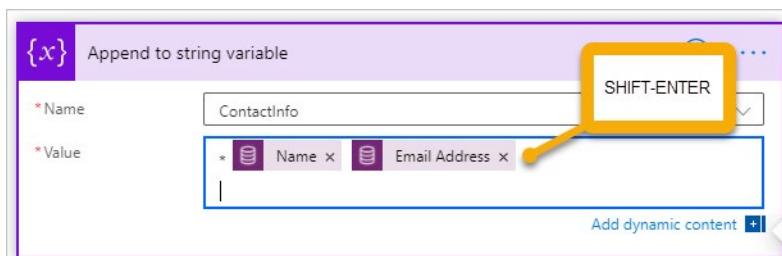
14. Select the **Insert new step** icon after **List rows**. Select **Add an action**.

15. In the search box, enter **Control** and then, under **Actions**, select **Apply to each**.

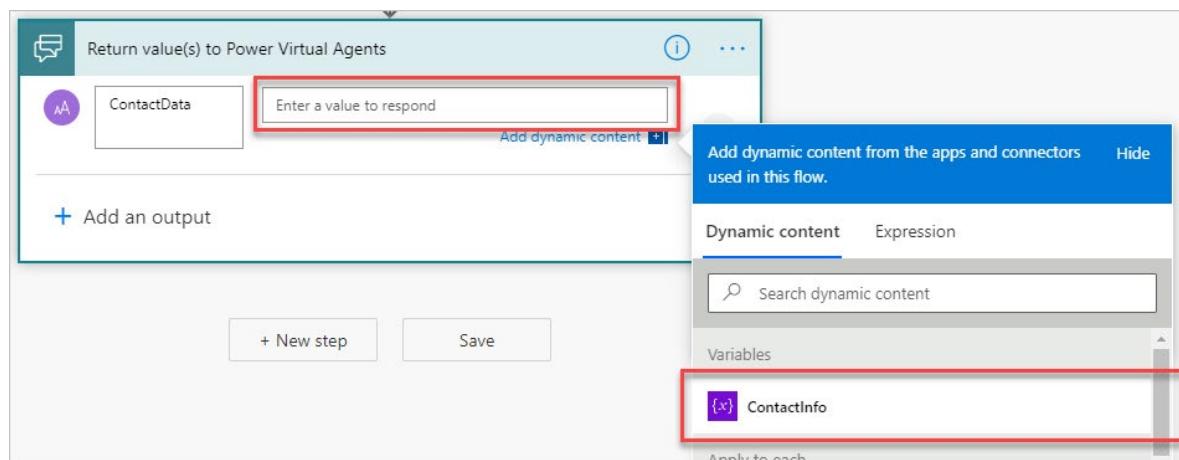
16. In the **Select an output from previous steps** field, select the text box and then select **value** under the **List rows** section. Select **Add an action** to continue.



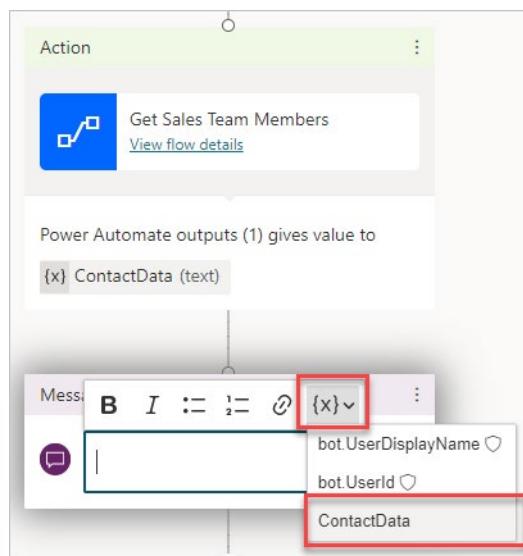
17. In the search box, enter **variable** and then, under **Actions**, select **Append to string variable**.
18. For the **Name**, select **ContactInfo** for the variable that you previously created in the flow.
19. Create a bulleted list with the name and email of each person. Power Automate uses Markdown syntax when formatting text. To indicate a bulleted list in Markdown, you need to use the asterisk \* at the line's start. To learn more about Markdown, see the Summary section at the end of this module.  
In the **Value** text box:
  1. Enter an asterisk \* and then add a space.
  2. Under dynamic content **List rows**, select **Name** and then add a space.
  3. Under dynamic content **List rows**, select **Email Address**.
  4. Press the **Shift + Enter** keys to start a new line.



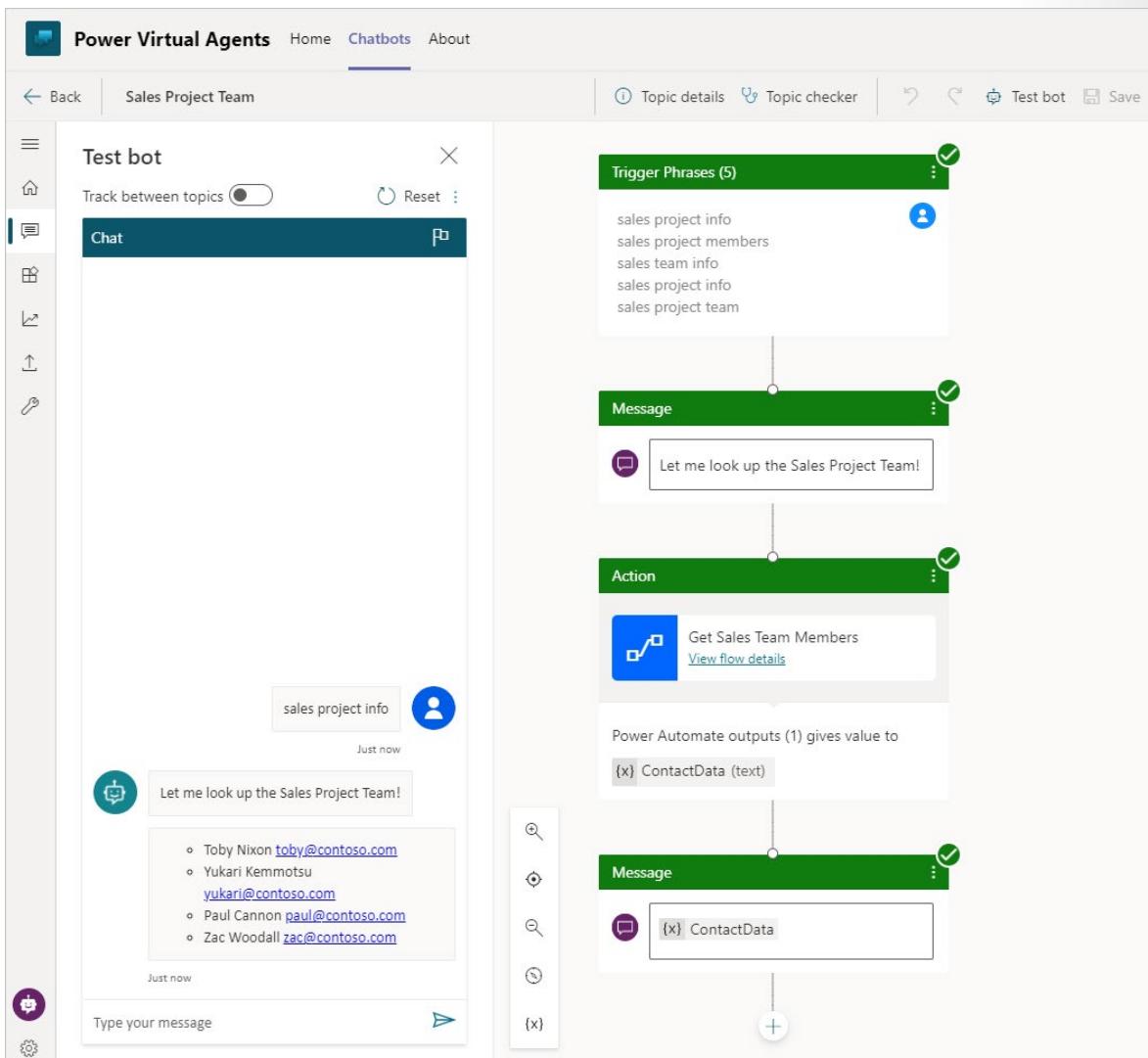
20. Select **Return value(s) to Power Virtual Agents** and then select **+ Add an output**.
21. Select **Text** as the type of output.
22. In the **Enter Title** text box, enter **ContactData**, which is the variable name that the chatbot used to receive the information.
23. For the **Enter value to respond** field, provide the variable that you have been using in the flow. Click in the text box and then select **ContactInfo** under the **Variables** section.



24. Select **Save** to finish editing the flow. Power Automate runs the **Flow Checker** to look for problems. You might see a warning about "...OData filter queries..." that can be ignored. Select **Close** to return to the chatbot editor.
25. Select the **Add node** icon after the **Message** node and then select **Call an action**.
26. The flow that you created will appear in the list. Select **Get Sales Team Members**.
27. Add a **Message** node after the **Action** node.
28. Click inside the text box to bring up the menu. Select **ContactData** under **Insert variable {X}**.



29. Select **Save**. If the **Test bot** pane is not visible, select the **Test bot** menu option. Enter one of the trigger phrases to test the chatbot.



Advanced flows are available to help you look up a single record in Dataverse for Teams based on the user's needs. For example, a user might want the contact information for a particular person based on their job title. While outside the scope of this module, Dataverse List Rows can use OData filters based on choices that the user has made in the chatbot. For more information regarding OData queries, see the Summary section at the end of this module.

## Publish and share your new chatbot

When the chatbot is ready, it needs to be published so that users can interact with it. Publishing is also required after you have edited the chatbot and you want users to be able to access the latest content. By having a published version, you can make future updates to the chatbot without impacting users.

The first time that the chatbot is published, only members of the team that the chatbot is associated with can access it. You can direct team members to the chatbot by providing them a link or by adding the chatbot to the **Built by your colleagues** section in Teams.

If the chatbot is for the whole organization, you will submit it for review by your Teams admin. After it has been approved, the chatbot will be accessible to all users and will appear in the **Built by your org** section in Teams.

## Publish the chatbot

To publish the chatbot, follow these steps:

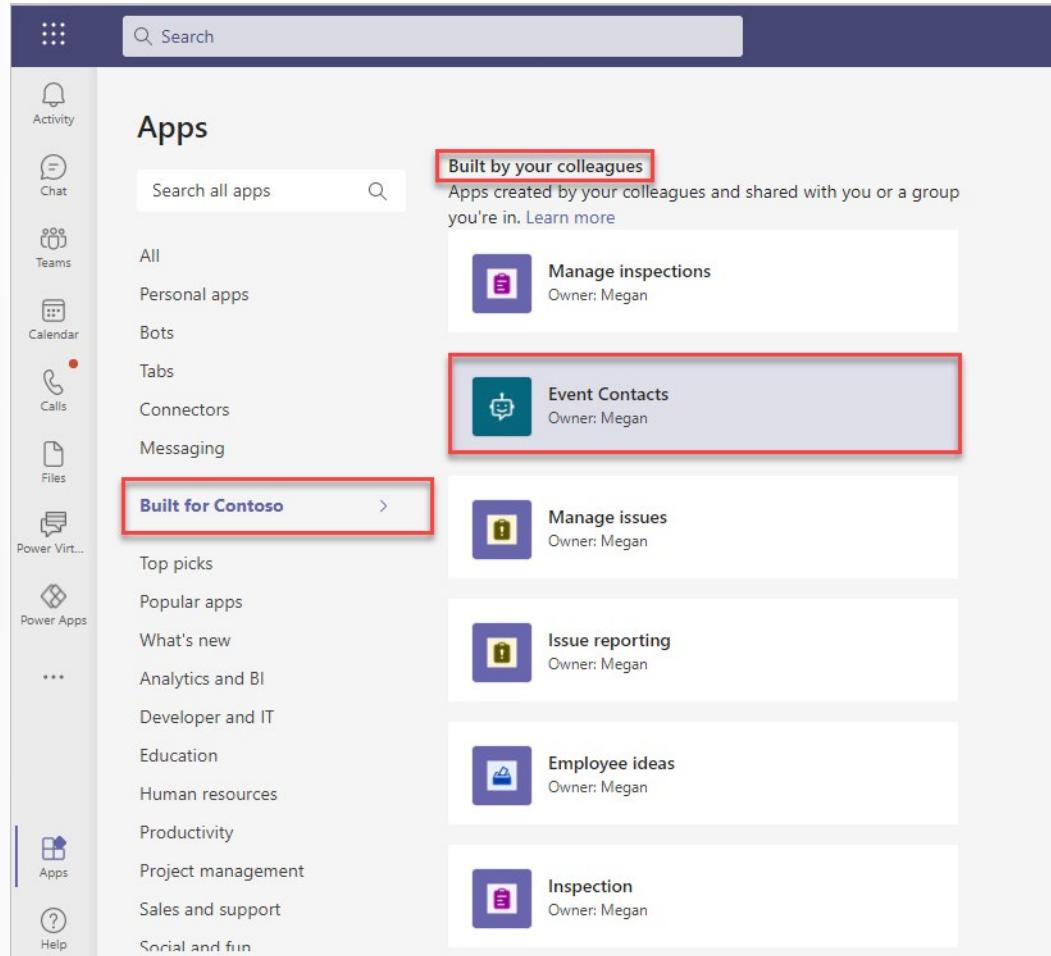
1. Open the chatbot for editing in the Power Virtual Agents app.
2. Select the **Publish bot** menu option or button.
3. Select the **Publish** button. In the **Publish latest content** confirmation window, select **Publish** to confirm.

Screenshot of the Publish latest content message.

If this is the first time that the chatbot is published, only team members can access it. If the chatbot is only intended for team members, you will need to direct users on how to access it. Select **Share the bot** to select the best method for your situation:

- **Copy link** - Send users a link to the chatbot.
- **Add to Teams** - Add the bot to the **Built by your colleagues** section under Teams Apps.

**Note:** Only team members can use the link or see the chatbot that is listed in the **Built by your colleagues** section in Teams.

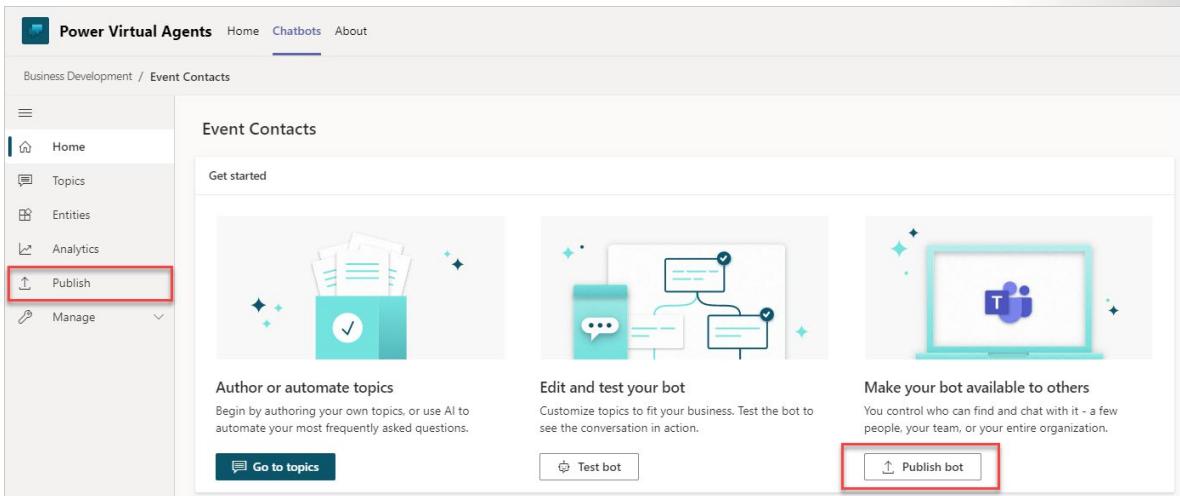


To make the chatbot accessible to the entire organization, continue on to the next section.

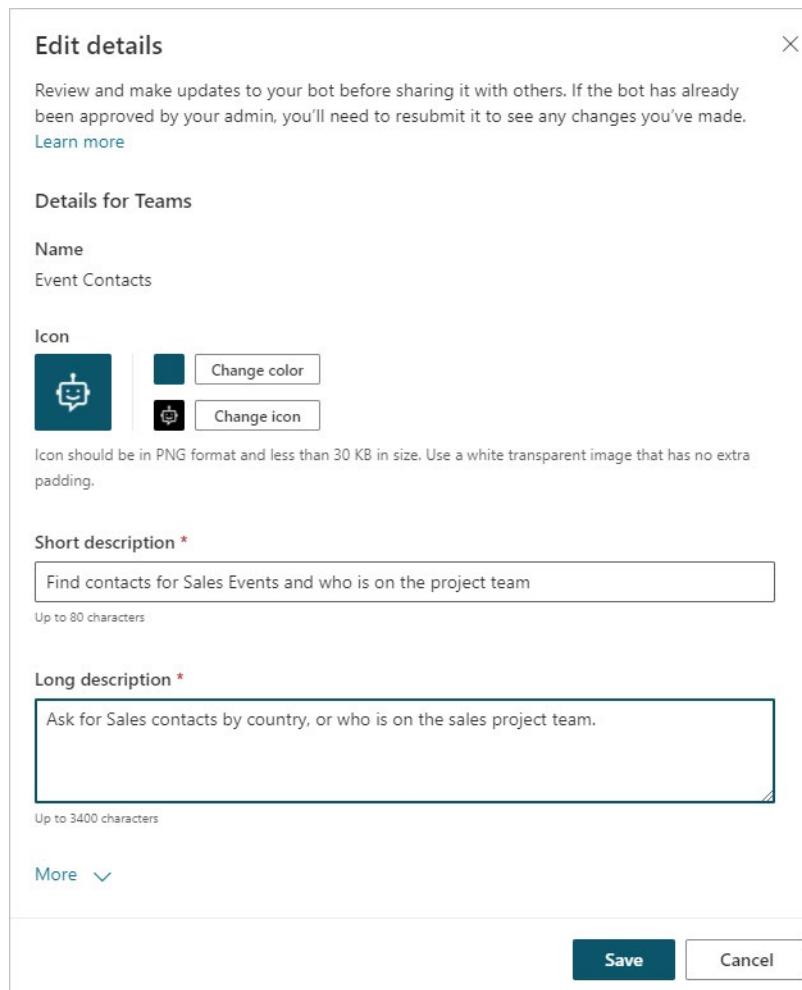
## Edit chatbot details

Before publishing a chatbot to the whole organization, you should provide more information to your users by editing the details that will appear in the **Built by your org** section in Teams.

1. Open the chatbot for editing in the Power Virtual Agents app.
2. Select the **Publish bot** menu option or button.



3. Select **Share the bot** and then **Edit details**.
4. From the **Edit details** pane, change the chatbot's icon, color, and description. Make sure that the description helps your users understand the information that the chatbot provides.



[!IMPORTANT] After a user has installed the bot from the Teams app store, changes to the bot icon and color will not apply to them. Users will have to uninstall and then reinstall the bot to see the changes.

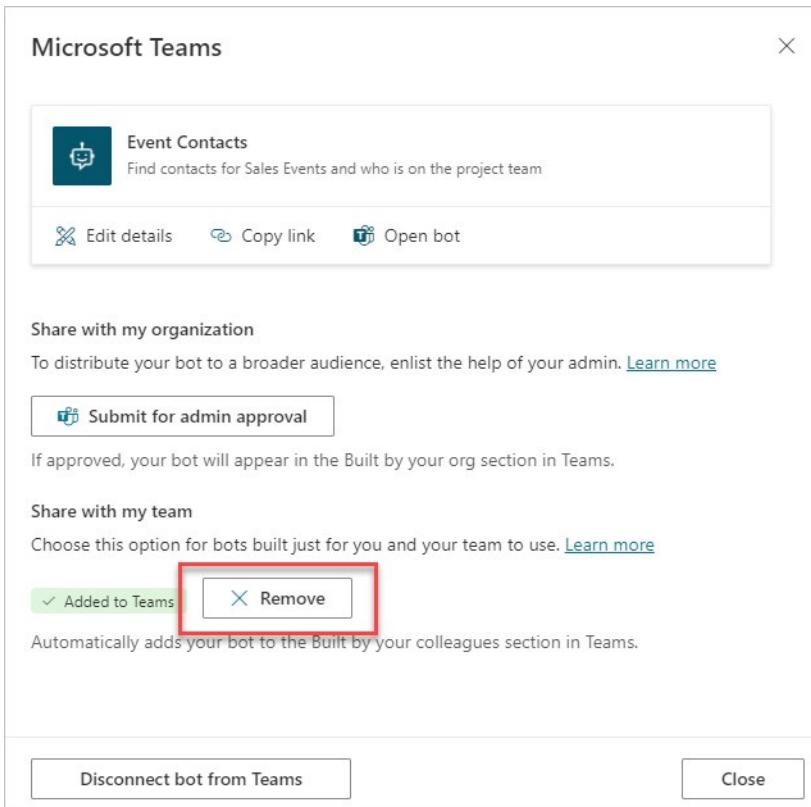
5. Select **More** to edit additional information like developer name and links to a website, privacy statement, and terms of use.
6. Select **Save** to close the **Edit details** pane.

## Publish the chatbot to the organization

For the entire organization to access the chatbot, a **request for admin approval** is needed. This process will send a request to your organization's Teams admin. After the request has been approved, all users will have access and the chatbot will appear in the **Built by your org** section in Teams. Any users can then discover and install the chatbot through the Teams app store.

To submit the request:

1. Open the chatbot for editing in the Power Virtual Agents app.
2. Select the **Publish bot** menu option or button and then select **Share the bot**.
3. Confirm that the app isn't already published to the app store for team members so they won't see the app twice in the Teams app store. Select **Remove** under **Share with my team** if it has been published.



4. Select **Submit for admin approval**.
5. Review the requirements for submission and then select **Submit for admin approval**.
6. Review the app details such as icon, color, and description, and then select **Submit for admin approval**.
7. A final confirmation window will appear. Select **Yes** to continue.

The request is sent to the Teams admin for approval. You can check the request's status by opening the **Publish** page and then selecting **Share the bot**. After the bot has been approved, it will appear in the **Built by your org** section of the Teams app store for everyone in your organization.

## Summary

In this module, you learned how to create a chatbot and define conversations with topics. You also learned how to use variables when asking users questions and call Power Automate to pull data from Dataverse from Teams. Additionally, you learned how to publish chatbots to your team or the whole company.