



Administering a low-code development platform - Power Apps and Power Automate Enterprise Deployment

Whitepaper

Summary: This is a technical whitepaper outlining considerations for planning, deploying and managing an enterprise Power Apps deployment.

Writers: David Yack (Colorado Technology Consultants)

Technical Contributors: Julie Yack (Colorado Technology Consultants), George Doubinski (solutions.NET), Kent Weare, John Landgrave, Manas Maheshwari, Jennifer Monroe, James Oleinik, Saurabh Pant, Imad Yanni, Denise Moran, Manuela Pichler, Carsten Groth, Saumil Shrivastava, Meera Mahabala, Manish Ojha, Darya Mazandarany, Per Mikkelsen, Anupma Sharma

Published: May 2020

CONTENTS

Introduction	3	Power Apps Build Tools (Preview) for Azure DEVOPS.....	94
Purpose of this whitepaper	3	Educate and Support	97
Scope of this Whitepaper	3	Hands on Labs.....	97
How to get started	4	Blogs	98
Next Steps	5	Community	98
Power Platform Overview	6	Support Ticket	98
Usage Scenarios	8	Submitting and Voting on Ideas.....	99
Platform Architecture	10	Microsoft Learn	99
Environments.....	10	Finding Consulting Partners.....	99
Common Data Service	13	Next Steps.....	100
Power Apps.....	15	Appendix	101
Power Automate	15	Appendix to environment strategy.....	101
Connectors	16	Appendix to resource sharing	102
On-premises Data Gateway	17	Appendix to On-premises data Gateway	108
Compliance and Data Privacy.....	17	Appendix to CDS security roles.....	110
Center of Excellence starter kit	19	Appendix to sharing apps in teams.....	114
Secure	21	Appendix to exporting apps and flows.....	119
Discovering your current state.....	21		
Licensing and License Management.....	33		
Layers of Security.....	38		
Monitor.....	52		
Working with the Admin Portals.....	53		
Log Power Apps Telemetry using Application Insights	63		
Power Apps and Power Automate Activity Logging via Microsoft 365.....	64		
Common Data Service Audit Logging	69		
Alert and Act.....	71		
Alert & Action via PowerShell or Power Automate leveraging Management connectors.....	71		
Deployment, ALM & Azure DevOps	81		
Solutions	81		
Application Lifecycle Management.....	87		

INTRODUCTION

Microsoft Power Platform is a high-productivity application development platform from Microsoft, it's a product family that delivers innovative business solutions across one seamlessly integrated platform. Power BI, Power Apps, Power Automate and Power Virtual Agents allow any business to analyze & visualize real-time business performance, quickly and easily build custom apps, automate workflows and integrate AI capabilities.

The platform is used by Microsoft to build their own 1st party applications Dynamics 365 Sales, Service, Field Service, Marketing and Talent. This means these applications are built natively on the platform. Enterprise customers can also build their own custom line of business applications using this same technology. Additionally, individual users and teams within your organization can build personal or team productivity applications with no-code or low-code.

PURPOSE OF THIS WHITEPAPER

This whitepaper is targeted toward the person or department responsible for planning, securing, deploying, and supporting applications built on the platform. The goal of the paper is to help you understand what is currently in your environment, how to proactively plan for applications being developed and deployed, and finally how to handle day-to-day administrative tasks to manage deployments.

In this whitepaper, we will cover key concepts, platform architecture, and decisions that will be necessary. Where possible we will help you develop best practices for your organization to ensure successful deployments and high productivity for users using the platform.

SCOPE OF THIS WHITEPAPER

Unless specifically noted, all features mentioned in this whitepaper are available as of April 2020.

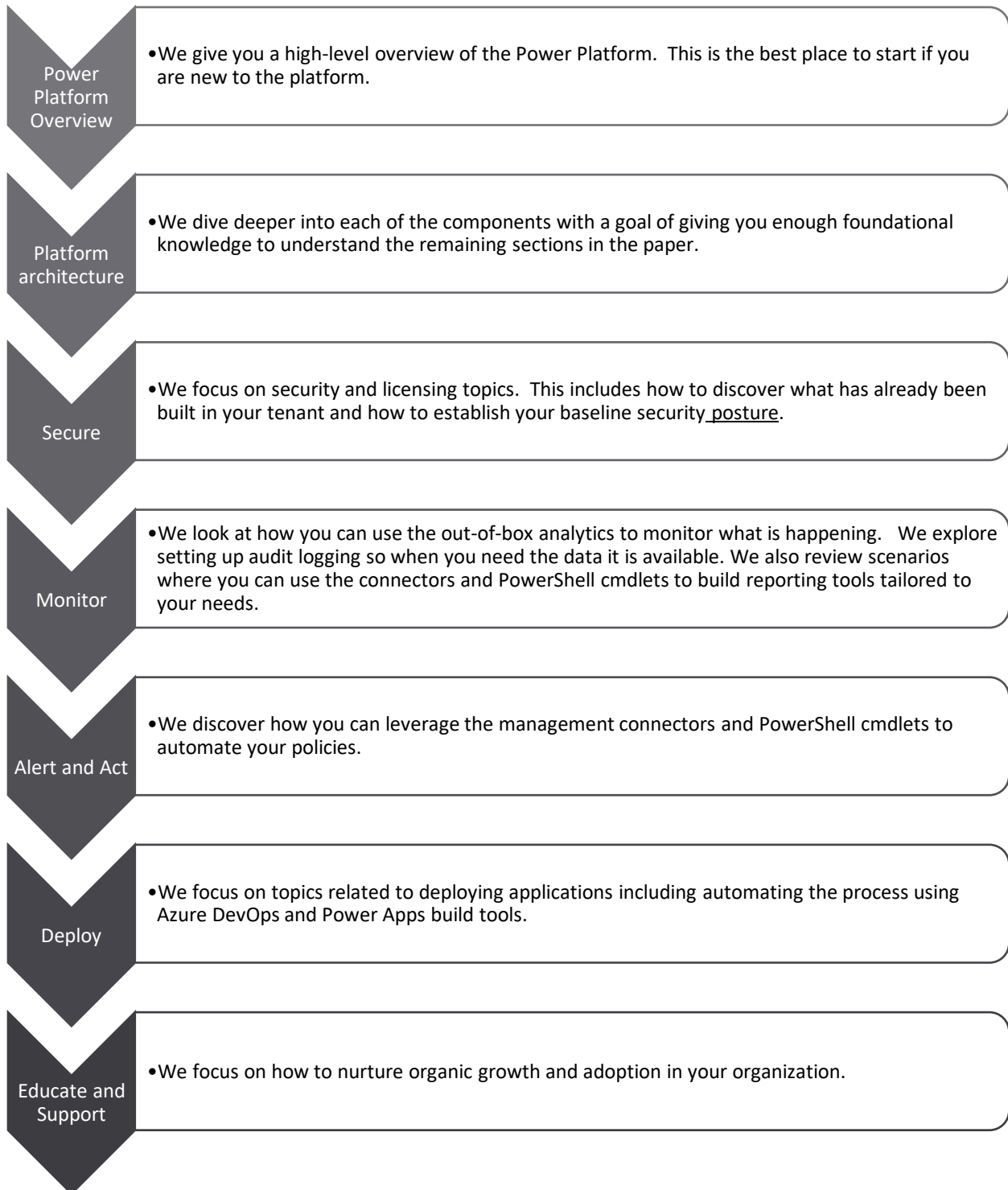
The following topics are out of scope for this whitepaper:

- Power BI and other parts of the broader Microsoft Power Platform
- Power Apps fundamentals for building applications
- ISV deployment scenarios, which are handled differently from enterprise deployment scenarios
- Performance tuning of applications
- Full deployment and management of 1st party Dynamics 365 applications
- Dynamics 365 Finance, Dynamics 365 Supply Chain Management, and Dynamics 365 Retail
- Third party solutions which integrate with Power Apps.

Please visit <https://docs.microsoft.com/en-us/power-platform/> to learn more about these topics.

HOW TO GET STARTED

While we recommend absorbing the whitepaper in its entirety, we thought it might be useful to give you some suggested areas on which to focus. We have organized this paper into the following sections. You can consume them in order or jump around as you wish.



NEXT STEPS

Following this whitepaper, your priorities should be

- Identify the central team that will be implementing Power Platform governance and assign them the [Power Platform service admin](#) role, which grants full access to Power Apps, Power Automate and Power BI
- Establish an environment strategy, restrict the creation of net-new trial and production environments to admins, and automate a process for requesting new environments
- Setup data loss prevention policies
- Leverage out-of-box activity logs & analytics
- Don't start from scratch, learn from the [Center of Excellence starter kit](#)
- Establish and automate your audit processes
- Welcome new makers and identify champions
- Establish a Center of Excellence that will help accelerate your adoption of the platform by investing in and nurturing organic growth while maintaining governance and control. Your Center of Excellence will be aligned to and drive your company's digital transformation strategy and goals

POWER PLATFORM OVERVIEW

Microsoft Power Platform is a product family that delivers innovative business solutions across one seamlessly integrated platform. Power BI, Power Apps, Power Automate and Power Virtual Agents allow any business to analyze & visualize real-time business performance, quickly and easily build custom apps, automate workflows and integrate AI capabilities.

Power Platform provides a low code interface for any user to quickly create custom apps while simultaneously providing robust tools for pro developers. This makes it possible to integrate innovative solutions across Azure, Modern Workplace, Dynamics 365 and standalone applications. At the intersection of these products lies digital transformation – giving the customer the power to innovate anywhere, while unlocking value everywhere.

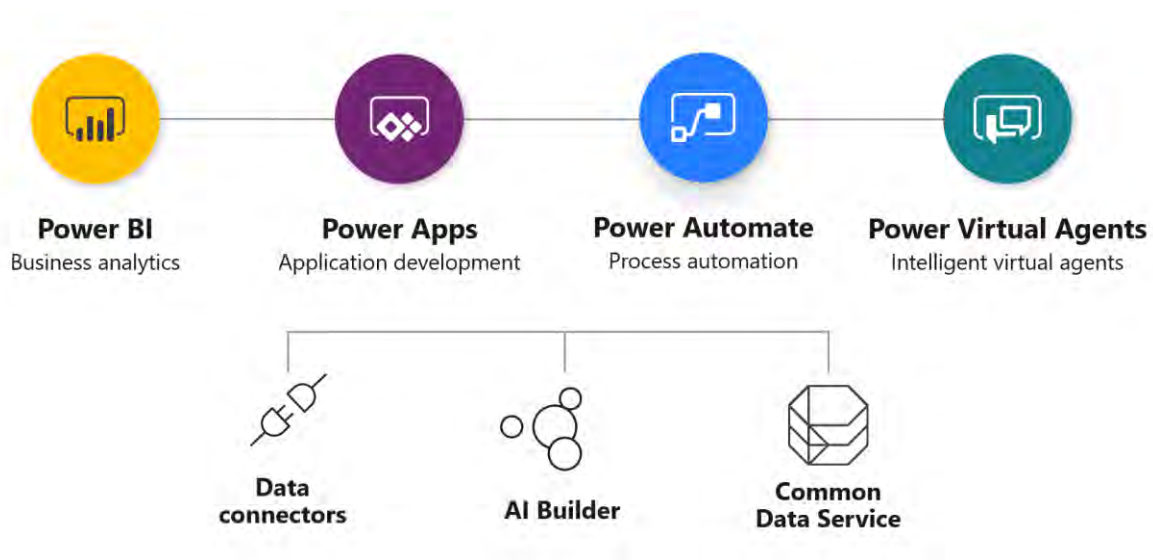


Figure 1 Power Platform Overview

The Power Platform includes several key concepts/components you should be aware of, for many of them we will dive deeper into as we progress forward in the whitepaper. Here are some of the key ones to get started:

<p>Power Apps</p>	<p>Power Apps is the toolset for low-code app development. There are three styles of applications; canvas, model-driven and portals. Power Apps canvas applications can also be embedded into SharePoint, Teams, Power BI and model-driven applications. Portals enable makers to build low-code, responsive websites which allow external users to interact with the data stored in the Common Data Service.</p>
<p>Power Automate</p>	<p>Automated workflows that orchestrate across services using connectors. Automations can be triggered to run when events occur in other systems and services or scheduled to run at a specific time. Users can start instant flows on demand from within the mobile app on the go or from the context of selected items in other apps. Business process flows can be</p>

	used to help guide users through a process. UI flows add robotic process automation (RPA) to automate repetitive tasks performed through a browser or user interface of a Windows app without use of a connector. Automations can quickly be created from templates or designed in Microsoft Visio, or by starting from blank in the portal designer.
Power BI	Power BI offers capabilities to help you discover and explore insights from your data including automated machine learning for predictive modelling, new AI visualizations with decomposition trees for detailed root cause analysis drill downs, and easier ways for everyone to interact with their data using the new Q&A visualization.
Power Virtual Agents	Power Virtual Agents empowers teams to easily create powerful bots using a guided, no-code graphical interface without the need for data scientists or developers. It eliminates the gap between the subject matter experts and the development teams building the bots, and the long latency between teams recognizing an issue and updating the bot to address it.
Common Data Service (CDS)	A mature data platform to manage data used by business applications, CDS is the platform that sits under the Dynamics 365 Sales, Service and Marketing business applications and is offered as a service to the Platform. An initial schema is defined by the Common Data Model. CDS provides built-in capabilities for business rules, workflows, calculated and rollup fields and more. CDS secures data using a configurable security model, that offers hierarchical, row level and field level security and auditing capabilities.
Common Data Model	An open-sourced definition of standard entities that represent commonly used concepts and activities. Every CDS database starts with the entities defined as "core". Through industry accelerators the Common Data Model is extended for verticals like banking, healthcare and others. Application builders can add their own custom entities to support specific business scenarios.
Connectors	There are 300+ connectors that make it easy for application builders to connect to both Microsoft and 3 rd party services, from Dynamics 365 to Dropbox. The connectors allow canvas apps and flows to easily use API (application programming interfaces) services without any developer knowledge. Custom connectors can also be configured to allow use of APIs that aren't covered by the public connectors.
AI Builder	AI Builder is a platform capability that allows easily adding AI to Power Automate, Power Apps and Dynamics 1 st Party Apps. This enables makers to automate tasks and predict outcomes without having to involve data scientists.
On-Premises Data Gateway	On-premises data gateway allows Power Apps, Power Automate and Power BI to reach back to on-premises resources to support hybrid integration scenarios. The gateway leverages Azure Service Bus relay technology to securely allow access to on-premises resources.

Power Apps Component Framework (PCF)

Power Apps component framework empowers professional developers and app makers to create code components for model-driven apps and canvas apps (experimental preview) to provide an enhanced user experience for the users to view and work with data in forms, views, and dashboards.

USAGE SCENARIOS

Power Apps is a flexible platform and can be utilized in several different types of scenarios:

- 1 Individual/Team Productivity Applications**

With self-service scenarios, users are empowered to take their own ideas of how they can optimize what they do every day and express them in the form of a Power Apps app or Power Automate automation. These assets can be shared with other team members and when successful, be promoted to be broader enterprise assets. Previously, these scenarios were out of reach and required high cost development resources to succeed. As an enterprise administrator your role is to put in place the guardrails to foster healthy individual productivity while at the same time safeguarding sensitive business data and ensuring continuity when individuals leave your company.
- 2 Dynamics 365 Applications**

These 1st party Microsoft applications are built on and therefore deployed into Power Apps environments and utilize the Common Data Service (CDS) for data storage and core platform services. These applications are the quickest way to tackle common business scenarios like customer engagement, while still allowing tailoring to your company's individual requirements. Custom Power Apps and workflows can be built, embedded into, or extend Dynamics 365 applications even further.
- 3 Apps from AppSource**

In addition to Microsoft built apps, 3rd party ISVs can also build on the Power Platform and are found via the AppSource marketplace. These apps and virtual agent bots can install into your existing environments or into their own depending on your unique needs.
- 4 SharePoint, Power BI, Teams**

Power Apps canvas apps can also be embedded into the applications users already use. Often this increases user adoption because they don't have to learn a totally new application from what they are already using. Admins retain granular controls over the content that their users can access and deploy.

 - Canvas apps are the primary way to customize SharePoint Online list and document library forms. Canvas apps can also be embedded as a webpart in your modern SharePoint page.
 - Power Apps can be embedded as a tab in Microsoft Teams channels or conversations, canvas apps can also be added as standalone apps, which are

discoverable in the Teams app store, and can be added to the left-hand Teams app bar. More information on sharing apps in Teams can be found in the

Appendix to sharing apps in teams

- Power Automate flows can be integrated as a Power Automate bot within Microsoft Teams, flows can also post messages to Teams channels and users and trigger on channel messages. Flows can also be built (from template or blank) and executed from within Teams
- Canvas apps can be embedded as a visual in Power BI

As an administrator you will be enabling these experiences and ensuring users have the right permissions and policies to interact with the applications.

5 Mission critical line of business applications

Using the same tools and techniques Microsoft uses to build Dynamics 365 apps, enterprise customers can build their own line of business applications. These differ from the individual productivity scenario above in that they often solve broader more complex problems. These applications are also often built by dedicated teams tasked with implementing them. The teams typically follow a more defined process for building the application. As an enterprise administrator you will be helping them put in place the necessary Application Lifecycle Management (ALM) to facilitate development and day to day operations.

These are the key scenarios you will encounter, but not an exhaustive list as it is up to the capabilities and the creativity of your organization to determine how it leverages the platform. As an enterprise administrator, you will be putting in place the necessary licensing, policies and processes to ensure success of the teams. Many organizations formalize this in a Center of Excellence chartered with nurturing and evolving the organizations efforts.

PLATFORM ARCHITECTURE

In this section we want to start drilling down in more detail on the components that make up the Power Platform.

ENVIRONMENTS

Environments are containers that administrators use to manage apps, flows, connections, and other assets; along with permissions to allow organization users to use the resources.

- Environments are tied to a [geographic location](#) that is configured at the time the environment is created
- Environments can be used to target different audiences and/or for different purposes such as dev, test and production
- [Data Loss Prevention \(DLP\) policies](#) can be applied to individual environments or the tenant
- Every tenant has a Default environment where all licensed Power Apps and Power Automate users can create apps & flows
- Non-default environments can be [created by licensed Power Apps, Power Automate and Dynamics users](#). Creation can be [restricted to only global and service admins via a tenant setting](#)
- An environment can have one or zero Common Data Service instances.

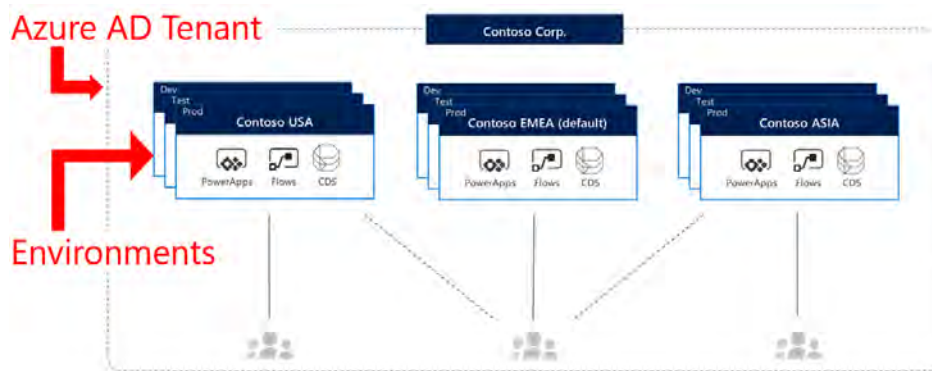


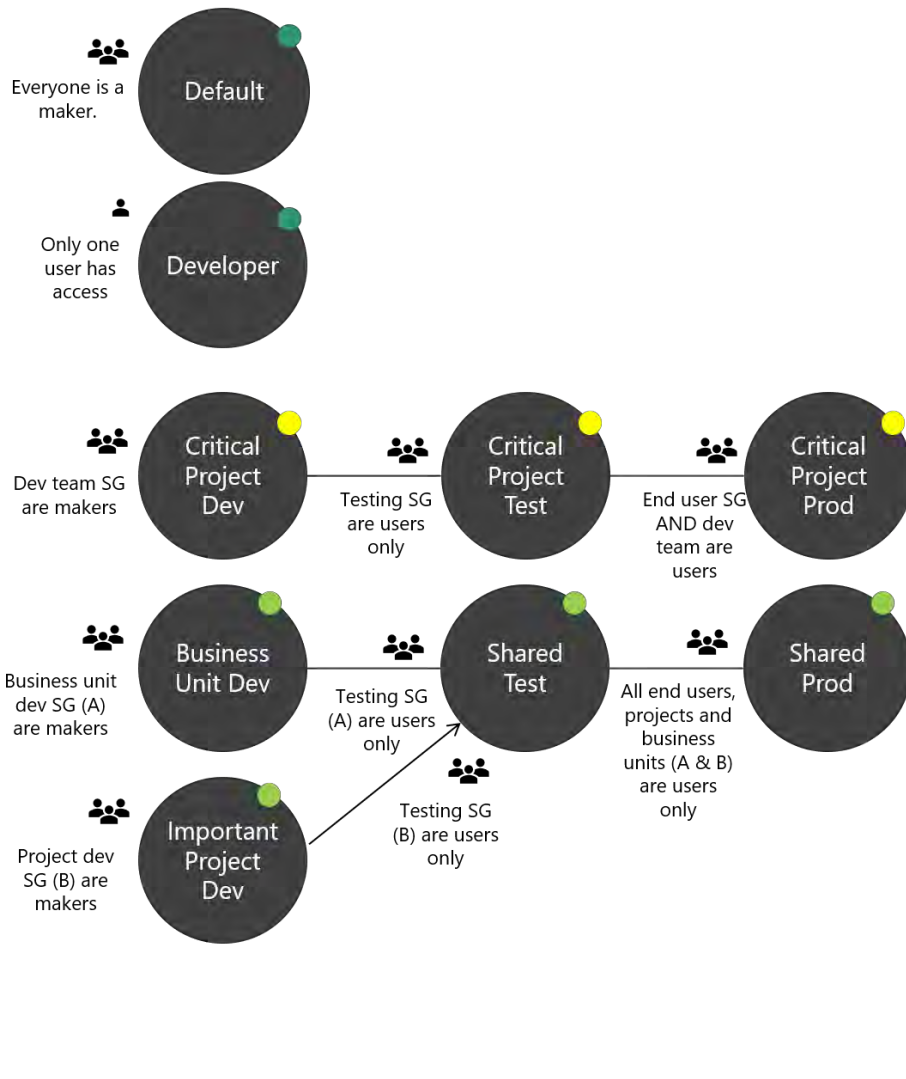
Figure 2 Environment Overview

DEVELOPING A STRATEGY

Developing an **environment strategy** means configuring environments and other layers of data security in a way that support productive development in your organization, while securing and organizing resources. A strategy to manage environment provisioning, access and controlling resources within them is important to:

- Secure data
- Understand how to use the Default environment correctly
- Manage the correct number of environments to avoid sprawl and conserve capacity

- Facilitate proper Application Lifecycle Management
- Organize resources in logical partitions
- Support Operations (& Helpdesk) in identifying apps that are in production by having them in dedicated environments
- Ensure data is being stored and transmitted in acceptable geographic regions (for performance and compliance reasons)
- Ensure isolation of applications being developed



Communicate with everyone that **Default** is not for development of critical apps.

Developer environments are completely locked for any other user except the user who subscribed to the community plan. Applications can be moved out of the environment if needed.

Dedicated dev/test/prod environments for each critical application. Developers have Environment Maker access in the dev environment, but only user access in test and prod. End users only have end user access to the production solution so no one can modify the applications.

Shared test/prod environments for important but medium complex apps can be shared between multiple projects or business units. Individual projects and business units should always have their own development environment to protect data. End users only have basic user access to solutions and data in production environments.

Figure 3: Example environment strategy diagram

Here are the main goals **for your environment strategy**:

<p>1</p> <p>Assign your admins the Dynamics 365 service admin role or Power Platform service admin role</p>	<p>2</p> <p><u>Restrict the creation</u> of net-new trial and production environments to admins</p>	<p>3</p> <p>Treat the default environment as a 'Personal productivity' environment for your business groups and consider renaming it</p>
<p>4</p> <p>Establish a process for requesting access or creation of environments</p>	<p>5</p> <p>Dev/Test/Production environments for specific business groups or application</p>	<p>6</p> <p>Individual-use environments for POCs and trainings</p>

Exploring these goals in more detail:

- **Assign your admins the Power Platform service admin or Dynamics 365 service admin role.**
 These roles provide administrative access to Power Apps canvas apps, Power Automate, model-driven apps, environments, custom connectors, connections, gateways, Power Portals, AI Builder models and all Common Data Service instances. This role should be assigned to admins that do not need global tenant admin access and are dedicated to managing Power Platform products. You can read a full comparison of the different admin roles. [here](#).
- **Restrict the creation of net-new trial and production environments to admins**
 Limiting environment creation is beneficial to maintain control in general: both to prevent unaccounted capacity consumption and to reduce the number of environments to manage. If users have to request environments from central IT, it's easier to see what people are working on if admins are the gatekeeper.
- **Treat the default environment as a 'Personal productivity' environment for your business groups.**
 Renaming the environment through the admin center is recommended to make the purpose of that environment self-explanatory. Clearly communicate that Default is never used for production apps, but for personal apps that aren't meant to be used by many.
- **Establish a process for requesting access or creation of environments**
 With environment creation locked down and default reserved for specifically personal apps, make it clear to developers in your organization that a proper development project should be started by requesting a new dedicated environment where there's clear communication of intent and support between developers and admins. In the next section there's more detail about automated environment creation, which is just one way to implement an easy formal request process.
- **Dev/Test/Production environments for specific business groups or application**
 In a perfect scenario without constraints, each project or business unit should have their own set

of development, test and production environments to develop solutions. Having staged environments ensures that changes during development does not break the end users in production and data is not corrupted. When resources are limited, focus this pattern for mission critical and important apps, or on business units who need their own dedicated space most

- **Individual-use environments for Proof of Concepts and training workshops**

To host workshops, hackathons and internal training events like App in a Day or Flow in a Day, create a new separate environment for the event to keep everyone organized. Ask the users to save the resources they need in a short term after the event and clean up the environment or reset it for other events.

COMMON DATA SERVICE

The Common Data Service (CDS) is a mature data platform used to securely store data for business applications built on Power Apps. CDS is an abstraction on top of underlying Azure cloud data management services to make it easier to build business applications. CDS provides not just data storage, but a way to implement business logic that enforces business rules and automation against the data. Data in CDS is organized as entities, for example account and contact would be two examples of entities. These entities can have relationships that define the business connection between the data stored in an entity. For example, John is the primary contact for Contoso would be expressed as a relationship from account to contact. The security model of CDS enables data protection down to the field level on individual records. A more thorough discussion of security will be covered in the [security section](#) of this paper.

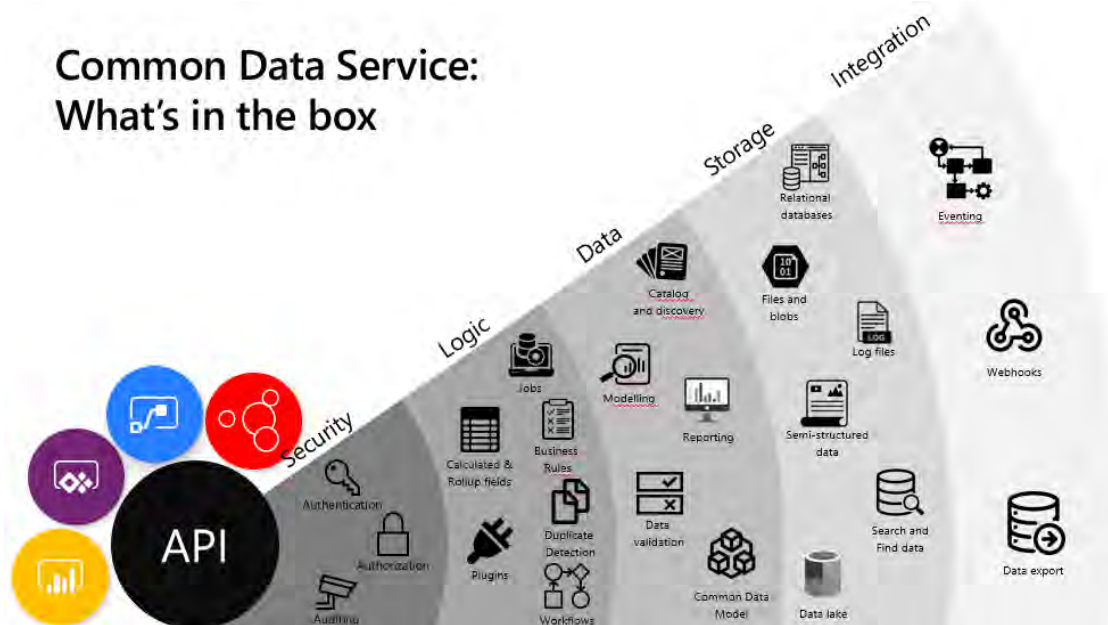


Figure 4 Common Data Service - What's in the box

CDS databases are created in the context of a Power Platform environment. Each environment can have only a single CDS database. CDS databases can be provisioned by you or licensed individuals in your organization to support their custom applications. A CDS database is automatically provisioned when a Dynamics 365 Customer Engagement application is added to your tenant. The CDS language and currency settings cannot be changed afterwards, enterprise organizations that operate in multiple regions should align on a base language and base currency.

MANAGING CDS DATABASE INSTANCES

The easiest way to know if you have a CDS database associated with your environments is to click on the environment in the list in the [Power Platform Admin Center \(https://aka.ms/ppac\)](https://aka.ms/ppac). If you look at the detail page of the environment and you see the following, where it indicates the database version then CDS has been created.

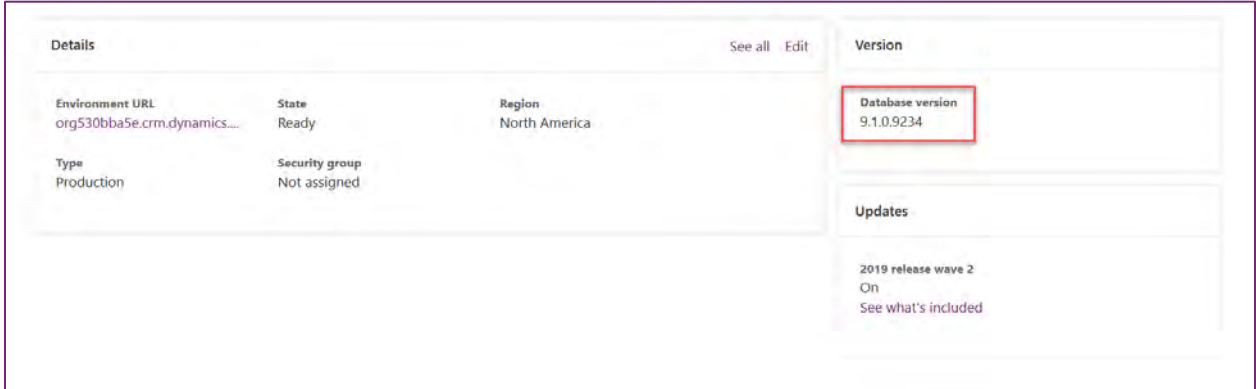


Figure 5 Power Platform Admin Center - Environment Details (Database version)

To create a database for an environment, click “+Add database” to create a database.

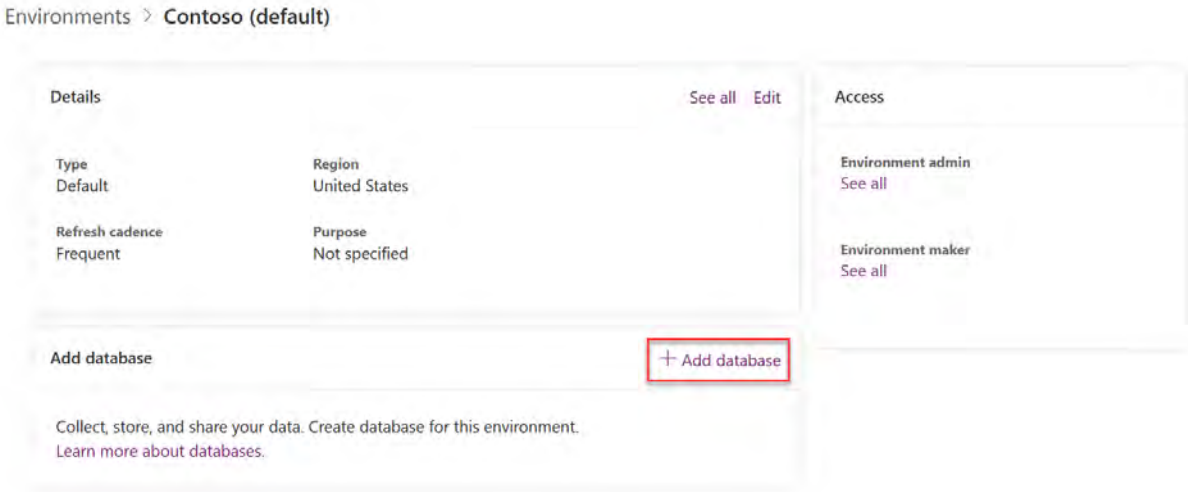


Figure 6 Power Platform Admin Center – Create CDS Database

Once CDS has been provisioned, from the details page you can also edit the environment URL and change the type of environment. For example, you could change a production environment to sandbox which would enable other features like copy and reset.

POWER APPS

There are three distinct types of applications: Power Apps canvas apps, Power Apps model-driven apps, and Power Apps portals. In this section, we will drill deeper into what you should know about these types.

Model-driven apps require a CDS database and are built on top of the data modeled in that database instance. Model-driven apps materialize views and detail screens based on the data structure. Model-driven apps are mobile friendly and in responsive layout out-of-the box and allow business process flow guidance, which in general leads to them being used for heavy data maintenance or backend operating teams

Canvas apps on the other hand can be built with or without a CDS database. They use connectors to access data and services. Canvas apps can start from a blank screen like an artist’s canvas and the creator manually lays out each screen. This allows the creator to have complete control of placement of controls on the canvas. Therefore, in many use cases they are for pixel-perfect designed applications that for instance are used by field workers from their mobile devices.

Power Apps portals are responsive websites that allow external users to interact with data stored in the CDS of the environment where the portal has been deployed. Allowing customers and partners to work with CDS data either anonymously or using commercial login providers including a variety of industry standard protocols like SAML2, OpenID Connect and WS-Fed. In order to use Power Apps portals a Common Data Service must be provisioned in the environment.

Regardless of the three types, apps will be built in the context of a Power Apps environment.

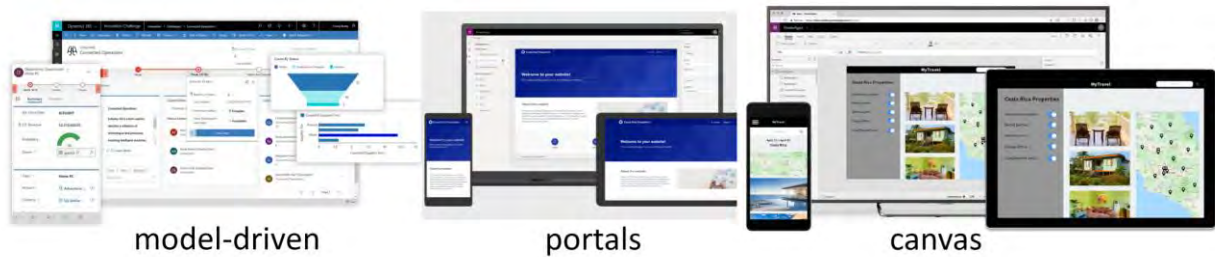


Figure 7 Power Apps Types

It is also possible as the use-cases get more complex that your solution contains multiple types of apps.

POWER AUTOMATE

Power Automate is an online workflow service that allows automating tasks across multiple services using connectors or UI automation.

Power Automate flows are started when a triggering event occurs, there are three types of triggers

- Automated (When an event occurs)
- Scheduled (Recurrence)
- Instant (Button click)

Once triggered, the flow proceeds to execute the actions defined in the flow. Conditions are used to guide the flow to the proper actions.

Business process flows (BPF) can be used to guide a user through an interactive process. For example, if you look at the Center of Excellence starter kit, it uses a BPF for reviewing compliance of Power Apps built by makers in the organization. Each stage in the BPF represents a milestone in the compliance review process. BPF can ensure data quality as it will guide users to the collection process of data in the context of current action.

Power Automate flows can be created

- From blank
- From a template
- From Visio

Flows created from a template can be modified and extended by the maker.

You can read more on how to use Visio here <https://docs.microsoft.com/power-automate/visio-flows>

The following is a simple example of a flow, that starts an approval when an item is created in a SharePoint list – this uses the SharePoint connector, Approval connector and Outlook 365 connector.

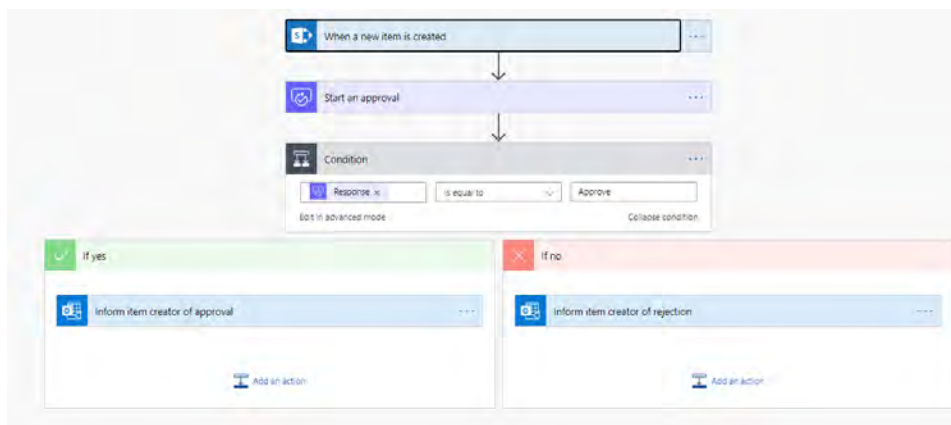


Figure 8 Sample Flow using SharePoint and Approval connectors

CONNECTORS

Connectors are essentially proxy wrappers around the application programming interfaces (APIs) provided by services that allow Power Automate, Power Apps and Logic Apps to easily interact with the service. Connectors can be either public or custom. There are currently over 300+ public connectors that can be used by all organizations. Examples of public connectors are Microsoft 365 (formerly Office 365), Common Data Service, Salesforce, SAP and more.

You may want to communicate with services that are not available as prebuilt connectors. [Custom connectors](#) address this scenario by allowing you to create (and even share) a connector with its own triggers and actions. Custom connectors are defined in the context of an environment and are only available to apps and flows within that environment. Connectors make triggers and actions available that can be used by the apps and flows. Triggers are used by Power Automate to start the execution of the flow. Actions are used by apps and flows to perform a defined set of actions during execution.

ON-PREMISES DATA GATEWAY

The on-premises gateway allows Power Apps and Power Automate to reach back to on-premises resources to support hybrid integration scenarios. The gateway leverages Azure Service Bus relay technology to securely allow access to on-premises resources.

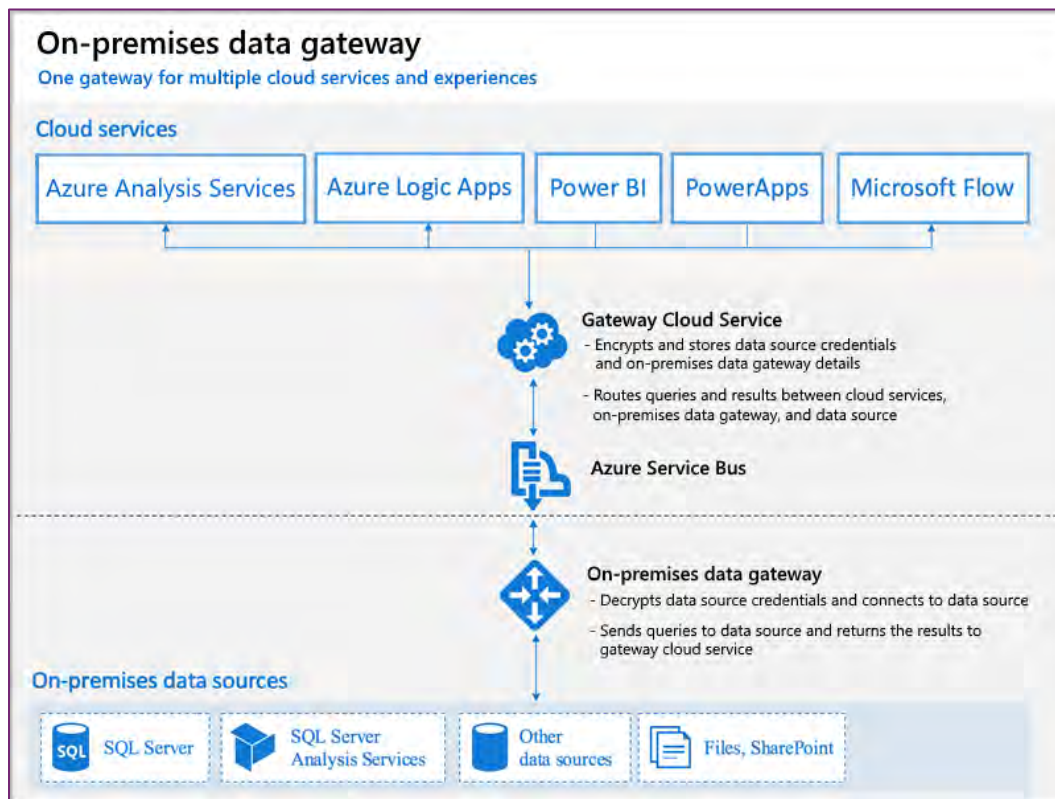


Figure 9 On-premises data gateway overview

COMPLIANCE AND DATA PRIVACY

Microsoft is committed to the highest levels of trust, transparency, standards conformance, and regulatory compliance. Microsoft's broad suite of cloud products and services are all built from the ground up to address the most rigorous security and privacy demands of our customers.

To help your organization comply with national, regional, and industry-specific requirements governing the collection and use of individuals' data, Microsoft provides the most comprehensive set of compliance offerings (including certifications and attestations) of any cloud service provider. There are also tools for administrators to support your organization's efforts. In this part of the document we will cover in more detail the resources available to help you determine and achieve your own organization requirements.

TRUST CENTER

The Microsoft Trust Center (<https://www.microsoft.com/trustcenter>) is a centralized resource for obtaining information on Microsoft's portfolio of products. This includes information on security, privacy, compliance, and transparency. While this paper may contain some subset of this information for the Power Platform, it is important to always refer to the Microsoft Trust Center for the most up to date authoritative information.

For quick reference, you can find the Trust Center Information for the Microsoft Power Platform here <https://www.microsoft.com/TrustCenter/CloudServices/business-application-platform/default.aspx> . This will include information on Power Apps, Power Automate and Power BI.

DATA LOCATION

Microsoft operates multiple data centers world-wide that support the Microsoft Power Platform applications. When your organization establishes a tenant, it establishes the default geographical (geo) location. In addition, when creating environments to support applications and contain CDS data the environments can be targeted for a specific geo. A current list of the geos for the Microsoft Power platform can be found here <https://www.microsoft.com/TrustCenter/CloudServices/business-application-platform/data-location> .

To support continuity of operations, Microsoft may replicate data to other regions within a geo, but the data will not move outside the geo to support data resiliency. This supports the ability to fail over or recover more rapidly in the event of a severe outage. There are some reasonable exceptions to keeping data in the specific geo that are listed on the above site primary focused on legal and support. It's also important to note, that you or your users can take actions that expose data outside of the geo. Other services can also be configured to access the data and expose it outside of the geo. By default, authorized users can access the platform and your applications and data from anywhere in the world where there is connectivity.

DATA PROTECTION

Data as it is in transit between user devices and the Microsoft datacenters are secured. Connections established between customers and Microsoft datacenters are encrypted, and all public endpoints are secured using industry-standard TLS. TLS effectively establishes a security-enhanced browser to server connection to help ensure data confidentiality and integrity between desktops and datacenters. API access from the customer endpoint to the server is also similarly protected. Currently, TLS 1.2 (or higher) is required for accessing the server endpoints.

Data transferred through the on-premises data gateway is also encrypted. Data that users upload is typically sent to Azure Blob storage, and all metadata and artifacts for the system itself are stored in an Azure SQL database and Azure table storage.

All instances of the Common Data Service database use SQL Server Transparent Data Encryption (TDE) to perform real-time encryption of data when written to disk, also known as encryption at rest.

By default, Microsoft stores and manages the database encryption keys for your instances, so you don't have to. The manage keys feature in the Power Platform admin center gives administrators the ability to self-manage the database encryption keys that are associated with instances of the Common Data Service. You can read more about managing your own keys [here](#) but generally, it is recommended have Microsoft manage the keys unless you have a specific business need to maintain your own.

CENTER OF EXCELLENCE STARTER KIT

A Center of Excellence is a coordinating function which ensures that change initiatives are delivered consistently and well, through standard processes and competent staff.¹ Establishing a Microsoft Power Platform CoE means investing in and nurturing organic growth while maintaining governance and control. A CoE is designed to drive innovation and improvement, and through its central function can break down geographic and organizational silos in order to bring together like minded people with similar business goals to share knowledge and success, experiment and encourage each other, whilst at the same time providing standards, consistency and governance to the organization. In summary, a CoE can be a powerful way for an organization to align around business goals rather than individual department metrics.

Establishing a Center of Excellence is not a requirement to be effective with the Power Platform. However, having one has had a positive effect on adoption and ease of administration at many organizations.

To help support those efforts Microsoft has created the [Power Platform Center of Excellence \(CoE\) starter kit](#). The starter kit is a collection of components and tools that are designed to help get started with developing a strategy for adopting and supporting the Power Platform, with a focus on Power Apps and Power Automate.

¹ Stephen Jenner and Craig Kilford, in Management of Portfolios

Download the most updated assets from the GitHub repository aka.ms/coestarterkitrepo. View the documentation for the Starter Kit here: <https://docs.microsoft.com/en-us/power-platform/guidance/coe/starter-kit>

The kit provides some automation and tooling to help teams build monitoring and automation necessary to support a CoE. The foundation of the kit is a Common Data Service (CDS) data model and Power Automate flows to collect resource information across the environments in the tenant. The kit includes multiple Power Apps and Power BI analytics to view and interact with the data collected. The kit also provides a number of assets that provide templates and suggested patterns and practices for implementing CoE efforts.

Throughout this paper we will highlight where the CoE starter kit has resources that can help jumpstart your efforts.

SECURE

In this section, we will focus on typical tasks as well as best practices for securing the platform. We will walk through

- Product discovery – understanding the origin of apps and flows
- Knowing your environments
- Layers of security
- Setting up Data Loss Prevention policies

DISCOVERING YOUR CURRENT STATE

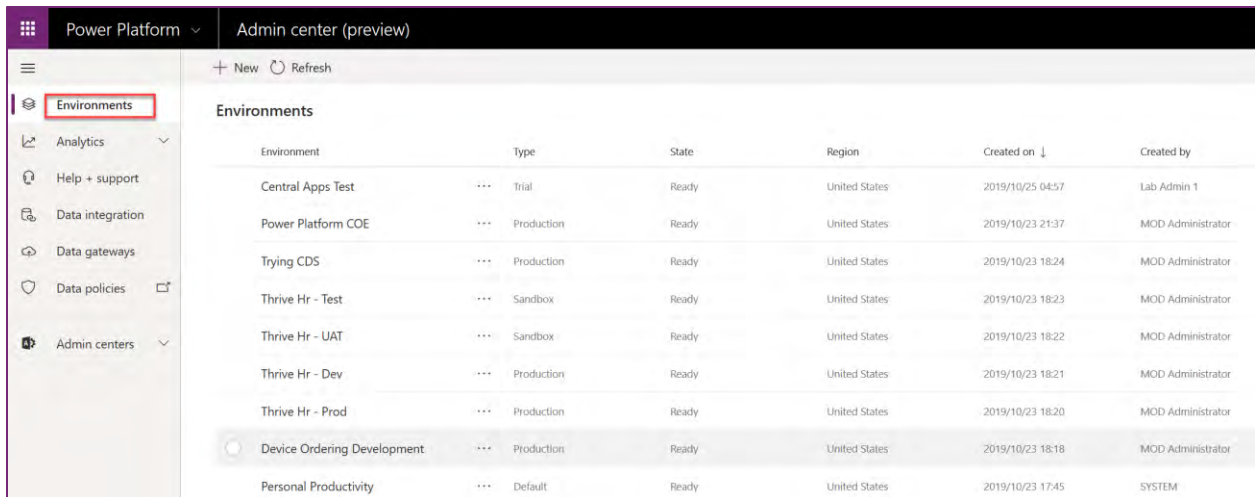
The first action you should take as part of securing your tenant is a quick discovery process to understand if your organization already has started to use Power Platform in your tenant. As part of this you should review what environments already exist and how makers have been building apps and flows.

Apps, flows and environments could already exist because

- Power Apps and Power Automate provide customization for Microsoft 365 (formerly Office 365) and Dynamics 365, and are included in this license
- Individuals can sign-up to learn and test out Power Apps through the Power Apps community plan
- Power Apps and Power Automate paid licenses give users the ability to build stand-alone apps and flows
- Production and Trial environments can be created by users, unless disabled in the Admin center

WHAT ENVIRONMENTS EXIST?

From the Power Apps Admin Center (<https://aka.ms/ppac>) you can see the environments that already exist. Key things to look for is how many, who created them, and what type of environments you have.



Environment	Type	State	Region	Created on	Created by
Central Apps Test	Trial	Ready	United States	2019/10/25 04:57	Lab Admin 1
Power Platform COE	Production	Ready	United States	2019/10/23 21:37	MOD Administrator
Trying CDS	Production	Ready	United States	2019/10/23 18:24	MOD Administrator
Thrive Hr - Test	Sandbox	Ready	United States	2019/10/23 18:23	MOD Administrator
Thrive Hr - UAT	Sandbox	Ready	United States	2019/10/23 18:22	MOD Administrator
Thrive Hr - Dev	Production	Ready	United States	2019/10/23 18:21	MOD Administrator
Thrive Hr - Prod	Production	Ready	United States	2019/10/23 18:20	MOD Administrator
Device Ordering Development	Production	Ready	United States	2019/10/23 18:18	MOD Administrator
Personal Productivity	Default	Ready	United States	2019/10/23 17:45	SYSTEM

Figure 10 Power Platform Admin Center - Environment List

ENVIRONMENT SECURITY ROLES

Different personas have different levels of access, and each are represented in two different ways depending on if the environment has CDS provisioned. If the environment has CDS, permission is controlled through the CDS Security Role model. If CDS is not provisioned in an environment, the permissions are based on environment role assignments (Environment Admin or Environment Maker).

Persona	Details	Environment has CDS	Environment does not have CDS
Environment Admin	Can perform all administrative actions on an environment.	System Administrator (predefined) security role	Environment Admin role assignment
Environment Maker	Can create resources (e.g., apps and flows) in an environment but cannot make administrative actions on the environment itself. If CDS is provisioned, they can optionally be assigned maker access to the database.	Environment Maker (predefined) security role for Canvas and Power Automate. System Customizer (predefined) security role for Model/CDS customization.	Environment maker role assignment
End user	Can access assets like apps and flow buttons that are shared with them but cannot create assets themselves. Note that end users are not given permission to the environment itself, they're only shared access to the applications and database that are located in an environment.	Customized security role that provide access to assets in the environment (such as CDS and Model Driven apps). If using canvas apps, access is shared the same as non-CDS environments—at the app level. Custom security roles are created to support applications built in your organization. Custom security roles can also come with applications you install from AppSource or if your users sign up for Dynamics 365.	Users are shared access to the canvas app (no environment role assigned)

TYPES OF ENVIRONMENTS

Here is a summary of the different types of environments supported from a product standpoint with comments on intended purpose and security:

Type	Description	Security
<u>Trial</u>	<p>Expires after 30 days, limit 1 per user.</p> <p>Can be used for short-term development, testing and personal exploration of the product.</p> <p>Provisioning trial environments can be restricted to admins.</p>	Full control
<u>Developer</u>	<p>One per user, single access, community program https://aka.ms/powerappcommunityplan</p> <p>Developer environments cannot be shared with nor affect other users.</p> <p>They are not meant to be used as production environments for apps.</p> <p>Provisioning developer environments cannot be restricted unless through a support ticket.</p>	Only single user account with community plan has access. Access cannot be shared
<u>Default</u>	<p>Every tenant has one default environment.</p> <p>Mainly used for personal exploration and productivity, by extending Microsoft 365 (formerly Office 365) services.</p> <p>Default should not be used to host production apps.</p> <p>Create a DLP policy to limit data flow between trusted Microsoft connectors and 3rd party APIs in the default environment.</p> <p>You cannot block the automatic provisioning of the Default environment.</p>	Limited control – all licensed users ² are Environment Makers

² Users licensed for Power Apps, Power Automate, Microsoft 365 (formerly Office 365) and Dynamics 365 Online, stand-alone licenses, free and trial licenses.

<p>Sandbox</p>	<p>Non-production environment enables some additional options like copy and reset.</p> <p>Used for development and testing, separated from production.</p> <p>Requires 1GB of CDS database capacity to provision.</p> <p>Provisioning Sandbox environments can be restricted to admins (since production environment creation can be blocked), but conversion from production cannot be blocked.</p>	<p>Full control.</p> <p>Developers require Environment Maker access to create resources.</p> <p>If used for testing, only end user access is needed.</p>
<p>Production</p>	<p>Non-expiring full environment</p> <p>Used to host production solutions and internal POC/ training workshops.</p> <p>Requires 1GB of CDS database capacity to provision.</p> <p>Provisioning production environments can be restricted to admins</p>	<p>Full control.</p> <p>In production environments, restrict end-user access as much as possible, just enough use the application. Do not give anyone Maker access</p>

DEFAULT ENVIRONMENT

Each tenant will have a default environment created automatically in the region nearest the Azure Active Directory (Azure AD) tenant. The default environment has a few distinct properties:

1. This environment can't be disabled or deleted.
2. All tenant users are added automatically to the Environment Maker role for the default environment and can't be removed from that role.
3. Only Microsoft 365 (formerly Office 365) tenant global administrators, Dynamics 365 service administrators, and Power Platform service administrators are added Environment Administrator.
4. As the default environment is the preferred place for individual users to start off building personal productivity apps and flows, it is recommended to rename the default environment to "Personal Productivity (default)" or another appropriate name for your organization.

The Default environment should not be used to host production solutions. It's designed to be an open environment that allows users to extend Microsoft 365 (formerly Office 365) and trusted applications or to build personal productivity applications that don't affect many people. You can implement this idea by adding a DLP policy that only allows data flow between trusted first party connectors.

ENVIRONMENT REGIONS

When you create an environment, you will pick a [geographic location](#). The location cannot be changed after creation. Application components, including the CDS database will reside in that region. Generally, you will want to choose a location closest to the application users for that particular environment. If you are connecting to other existing external resources, you should consider their location as well. You should also consider any data residency concerns when choosing a location.

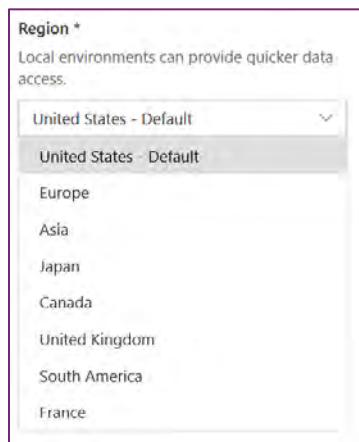


Figure 11 Environment Regions List

Additionally, there is a region named Preview (United States), this environment can currently only be created through the admin portal. Creating an environment in this region provides the ability to test portal and designer changes ahead of them rolling out to the normal regions. For example, new connectors or connector actions likely will be available here prior to release. It's important to note that as of this writing you can't create a CDS database in an environment in the preview region.

WHO CAN CREATE ENVIRONMENTS

As a global administrator or service administrator in the [Power Platform admin center](#) you will be able to see a list of all environments created in your tenant.

TRIAL ENVIRONMENTS

By default, one trial environment at a time can be created by paid and trial licensed users. These trial environments do not count against the tenant storage capacity and expire after 30 days if not converted

to production environments. It's important to understand if not converted prior to the 30 days all resources including data in that environment are deleted.

For more details on who can create environments review the docs here

<https://docs.microsoft.com/power-platform/admin/create-environment#who-can-create-environments>.

It is **recommended to restrict trial environment** creation to only Microsoft 365 (formerly Office 365) global admins, Dynamics 365 admins, and Power Platform admins from the [Power Platform admin center](#):

1. If you are a Microsoft 365 (formerly Office 365) global admin, Dynamics 365 admin, or Power Platform admin then navigate to the [Power Platform admin center](https://aka.ms/ppac) (<https://aka.ms/ppac>).
2. Select the settings "gear" in the header:



Figure 12 Settings in Power Platform Admin Center

3. Under 'Who can create trial environments' and select the **Only specific admins option** and select

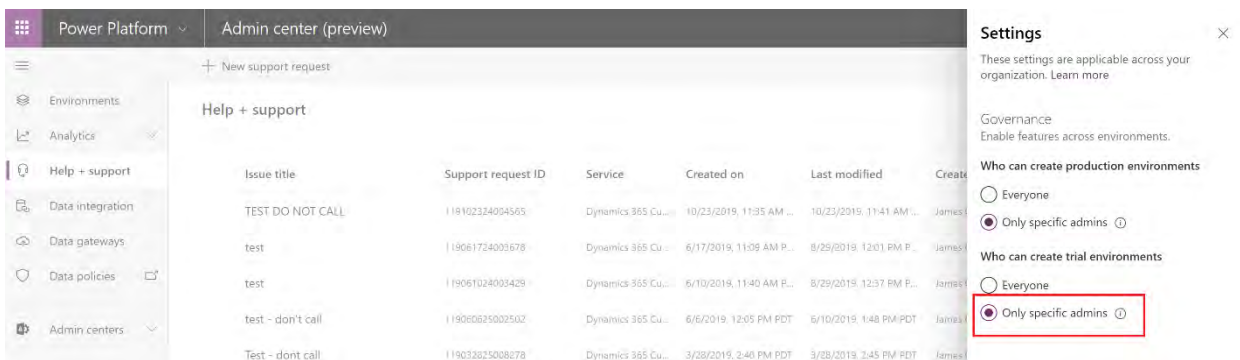


Figure 13 Governance Settings in Power Platform Admin Center

PRODUCTION ENVIRONMENTS

By default, all licensed users will be able to create new production environments if the tenant has at least 1GB of CDS database storage capacity remaining.

It is **recommended to restrict production environment** creation to only Microsoft 365 (formerly Office 365) Global admins, Dynamics 365 admins, and Power Platform admins from the [Power Platform admin center](#):

1. If you are an Microsoft 365 (formerly Office 365) Global admin, Dynamics 365 admin, or Power Platform admin then navigate to the [Power Platform admin center](https://aka.ms/ppac) (<https://aka.ms/ppac>).
2. Select the settings “gear” in the header:



Figure 14 Settings in Power Platform Admin Center

3. Under ‘Who can create production environments’ select the **Only specific admins option** and **select**

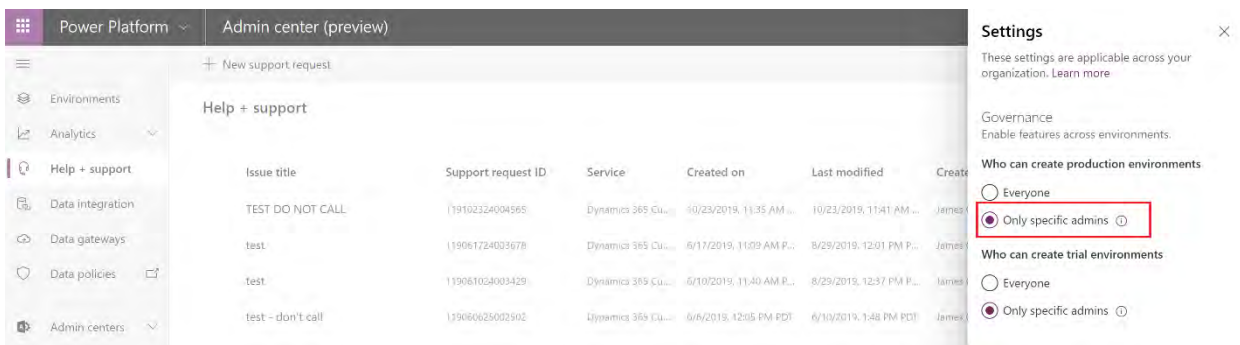


Figure 15 Governance Settings in Power Platform Admin Center

WHO CAN MANAGE ENVIRONMENTS?

Environment management is based on roles.

- Global admin/ Dynamics 365 service admin can manage any environment in the tenant.
- Licensed users need to have Environment administrator role to manage the environment.

No additional Power Apps / Power Automate plan license is required to manage environments.

Microsoft 365 (formerly Office 365) Global Admin	Dynamics 365 Service Admin Power Platform Service Admin	Delegated Admin
Full administration to all services in tenant	Full administration to all Power Apps and Power Automate assets and environments Power Platform Admin role	Full administration to all services in tenant Used for partners to provide support to customers

Full support for Power Apps and Power Automate coming soon

View the Service administration permission matrix here: <https://docs.microsoft.com/power-platform/admin/use-service-admin-role-manage-tenant>

This table will provide an overview of the different features and Environment admin vs a Global Admin / Dynamics 365 service admin has access to:

Feature	Environment Admin	Global Admin Dynamics 365 Service Admin Power Platform admin
View environments	*Yes	Yes
Create environments	**Yes	Yes
Edit environments	*Yes	Yes
Delete environments	*Yes	Yes
Backup	*Yes	Yes
Restore	*Yes	Yes
Reset	*Yes	Yes
Copy	*Yes	Yes
Resource/apps/solution	*Yes	Yes
Analytics – Common Data Service	*Yes	Yes
Analytics – Power Apps	*Yes	Yes
Analytics – Power Automate	*Yes	Yes
Analytics – Capacity	*Yes	Yes
View Help + Support tickets	Yes	Yes
Create Help + Support tickets	Yes	Yes
View data integration	*Yes	Yes
Create data integration	*Yes	Yes
View data loss prevention	*Yes	Yes
Create data loss prevention	*Yes	Yes

*Yes – Only applicable to environments, user is an Environment administrator

Yes – Have full access to the feature

**Note : Create environments feature is available to any licensed user in the tenant. If you would like to disable this capability and only allow Global Admins and Dynamics 365 Service admins access to this feature, run the PowerShell command described below

For example – If the feature is enabled, a user with an office license can also provision an environment. Any licensed user can create environment.

WHAT APPS AND FLOWS ALREADY EXIST?

From the Power Platform Admin Center at (<https://aka.ms/ppac>) → Environments you can look at the detail for each environment and from inside the Resources see a list of any apps and flows that are associated with that particular environment.

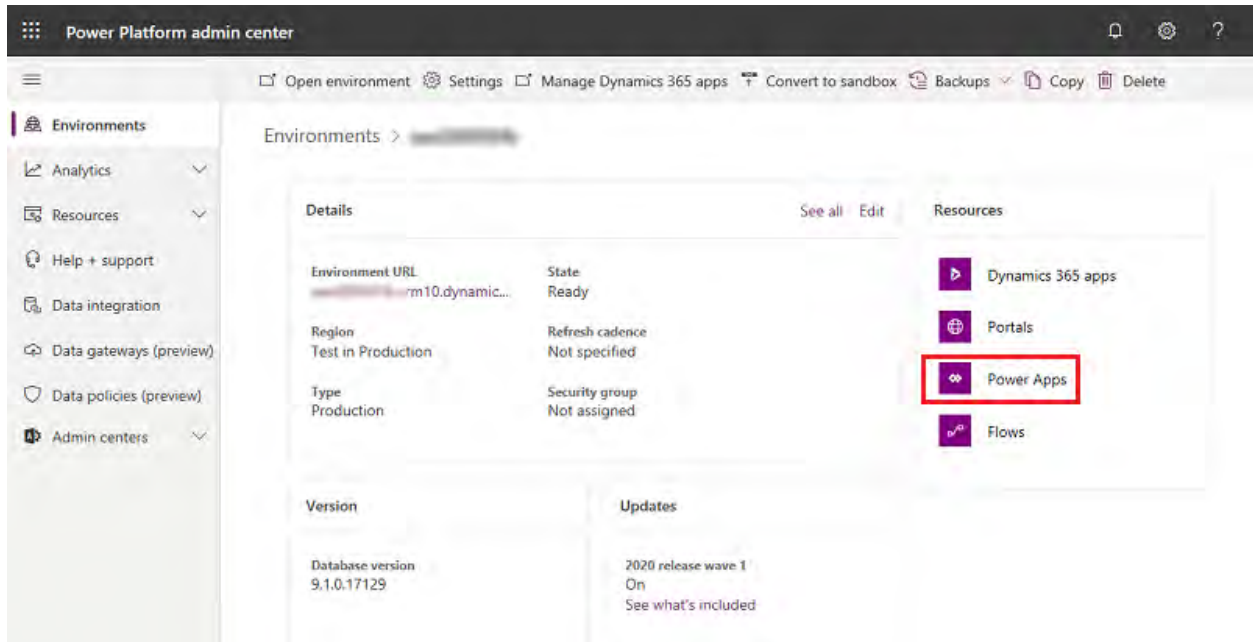


Figure 16 Resource List in Power Apps Admin Center

For Power Automate flows, you also have the ability as an administrator to turn on/off the flow as well as delete it. You can also manage sharing of the flow from here.

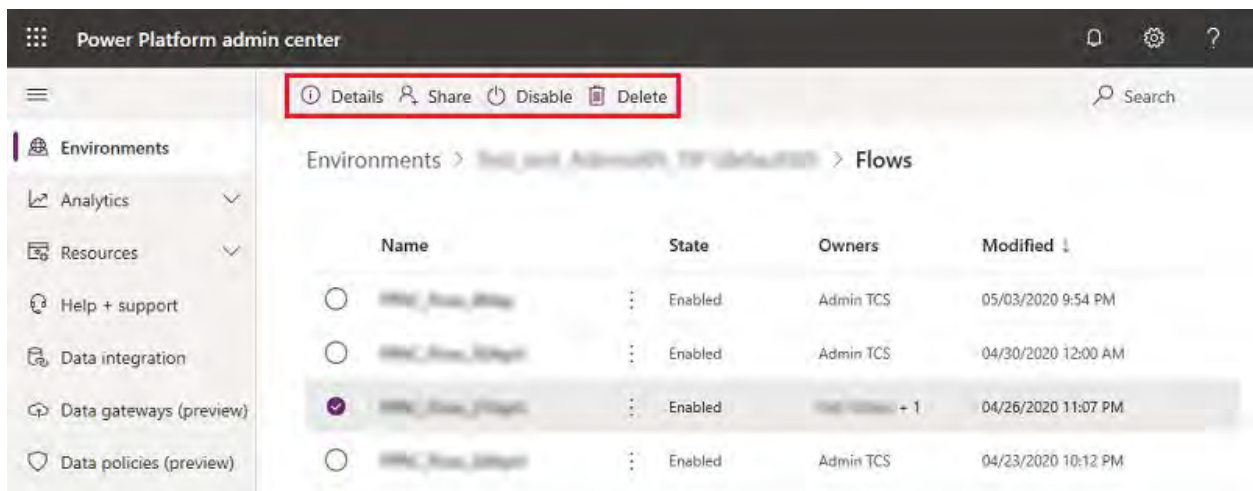


Figure 17 Flow Resource List

You can also explore this information from the PowerShell cmdlets – we cover more details on that in the **Alert and Act** section of this whitepaper.

The CoE Starter Kit (<https://aka.ms/coestarterkit>) also can be used to perform this discovery using the Power Platform Admin View app that is included. The advantages of the CoE app is it provides a tenant wide view of your resources. The following is an example of the app.

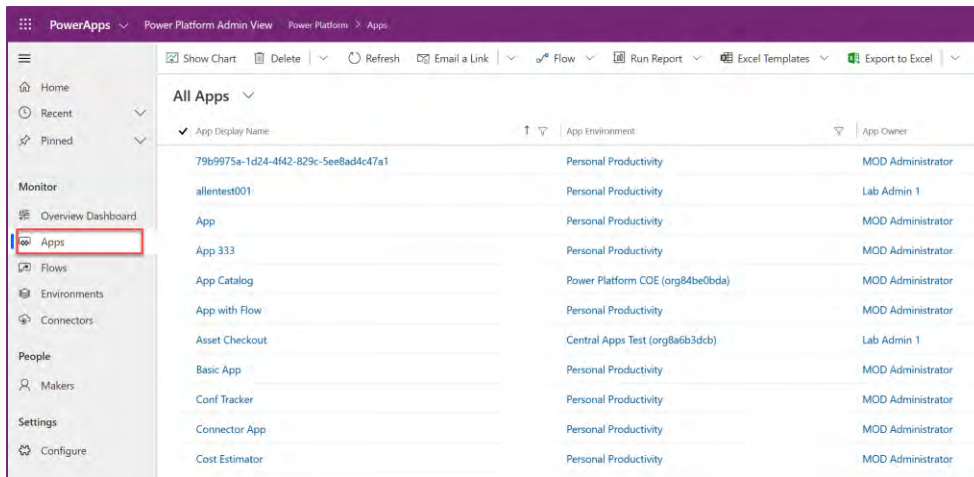


Figure 18 Admin View - CoE Starter Kit

In the default environment, you might see organic growth and adoption of the platform, and you will therefore see a mix of apps and flows here:

- Test, trial and training apps – users are following labs (like App and Flow in a day), trying out templates or testing features. These are experimental apps that are likely not shared with anyone.
- Productivity apps – these apps are used in production by the owner or a small group of people
- Critical apps – these apps have been created and are now critical to the operation of this business unit

As part of your discovery phase, you will identify and categorize the apps and flows already created. The out of the box analytics in aka.ms/ppac will help you discover App Usage by Unique Usage and Launches – navigate to aka.ms/ppac → Analytics → Power Apps

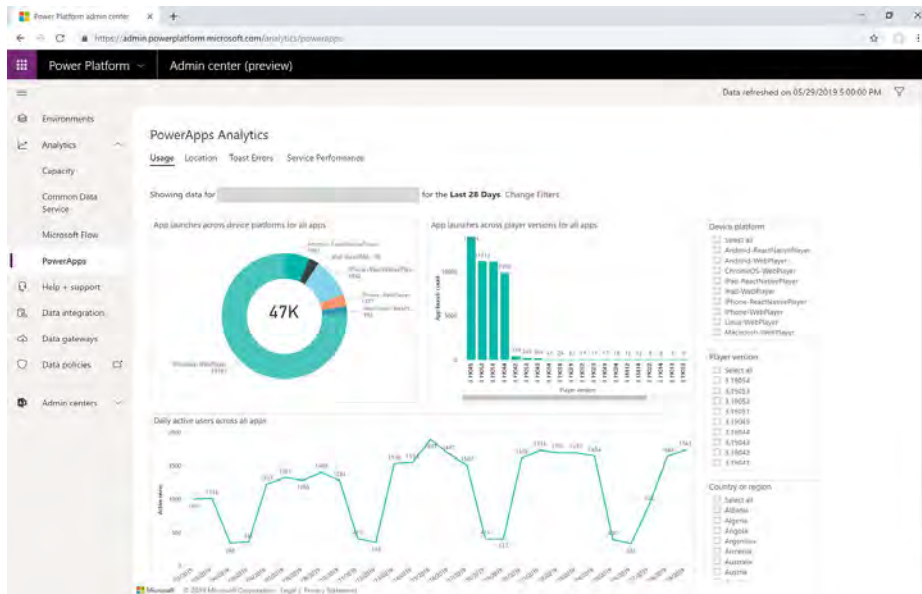


Figure 19 Power Apps Analytics

WHAT POLICIES DO WE ALREADY HAVE?

Data Loss Prevention policies (DLP policies) enforce rules for which connectors can be used together and which are blocked. Connectors are classified as Business Data only or No Business Data allowed or blocked. A connector in the business data only group can only be used with other connectors from that group in the same app or flow.

From the Power Platform Admin Center (<https://aka.ms/ppac>) → Data Policies you can see the current policies you have in place in your tenant. This should be your first stop as a new administrator to understand what policies are currently in place.

We explore DLP policies in more detail the **Data loss prevention policies** section of this whitepaper.

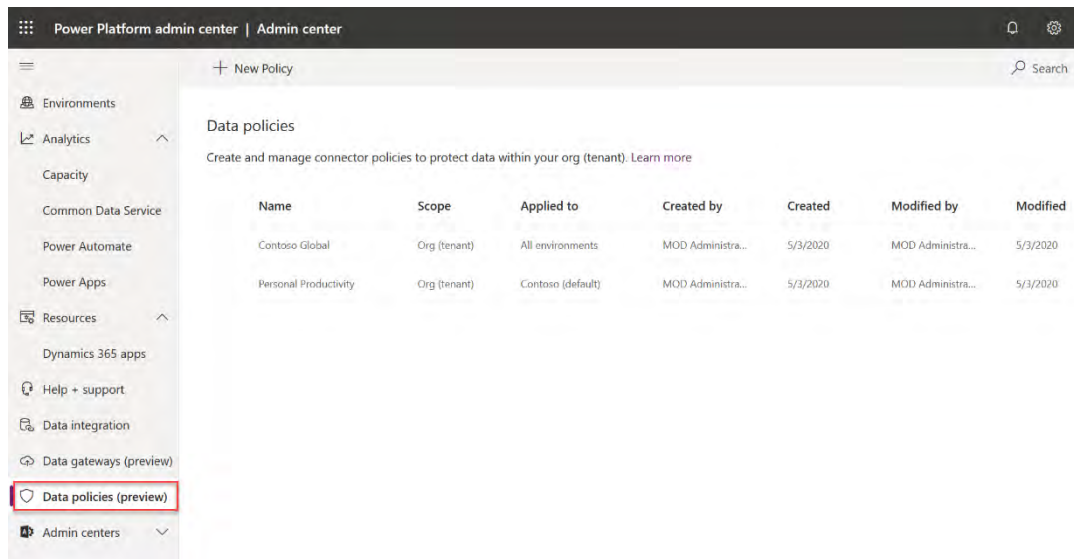


Figure 20 DLP Policies

The CoE starter kit implements a DLP Editor, using the [Power Platform Management Connectors](#). Through this app, you can see the impact of a DLP policy change, mitigate the risk and then update the policy.

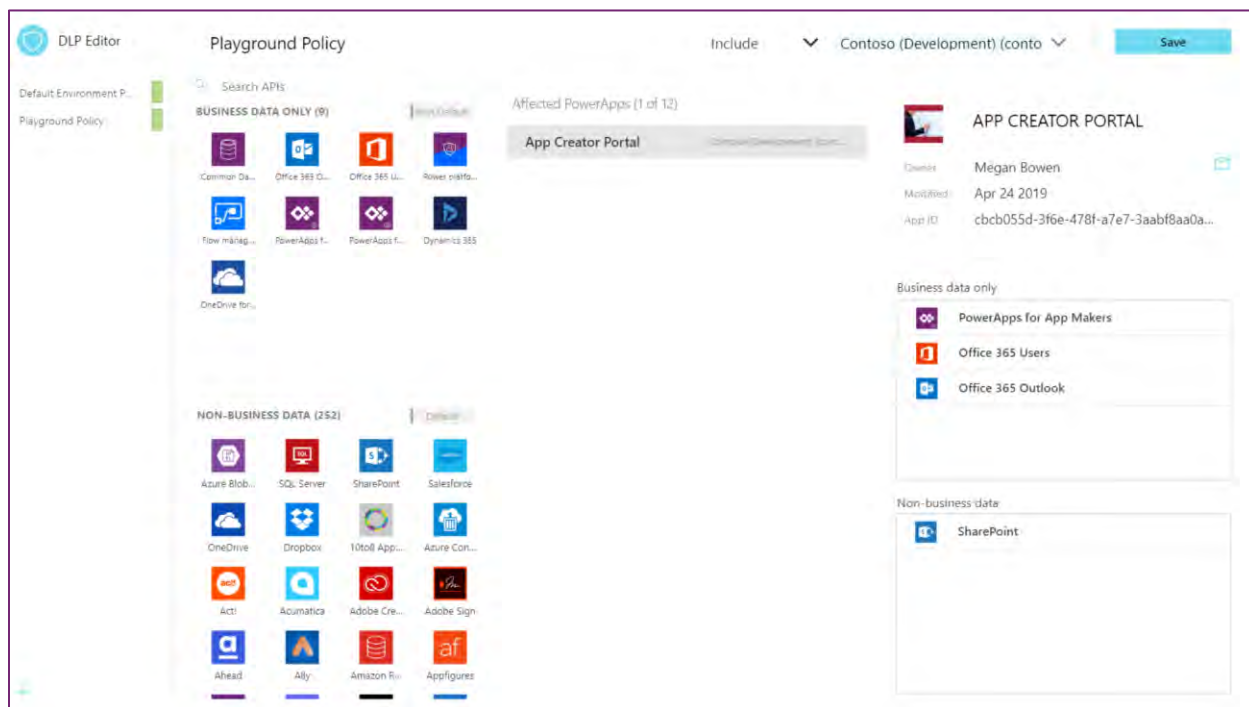


Figure 21 Screenshot of the DLP Editor (part of the CoE Starter Kit)

WHAT USERS ARE ALREADY LICENSED?

You can always look at individual user licensing in the Microsoft 365 (formerly Office 365) admin center by drilling into specific users. From the Power Apps administration center, you can also produce a report

focused on Power Apps licenses. This is one of the steps we recommend you do right away as a new administrator trying to understand your current licensing.

You can download the report from admin.powerapps.com → **Tenant** → **User Licensing**

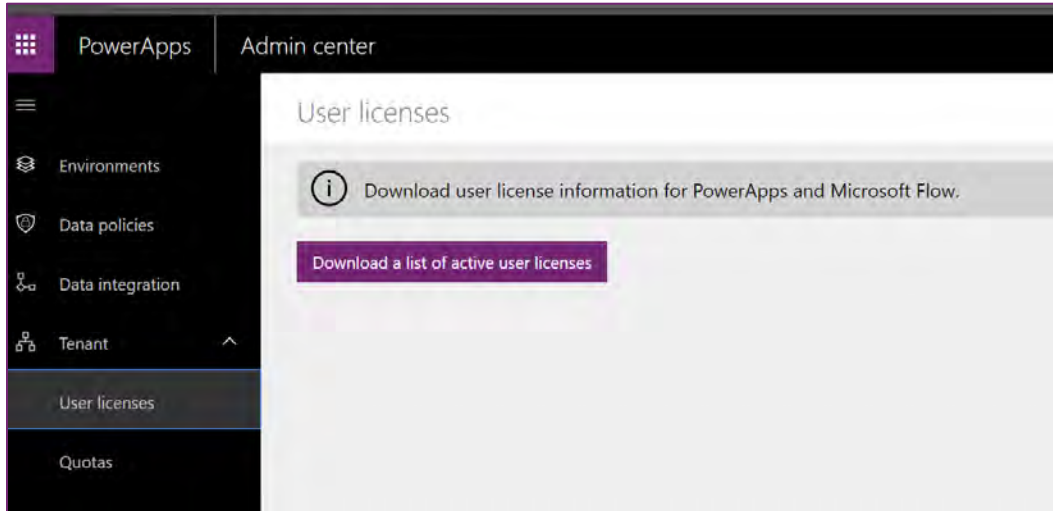


Figure 22 Download User licenses from Power Apps Admin Center

The report is an Excel workbook:

	A	B	C	D	E	F
1	User Name	Email Address	Service	License	License Assigned Date	Is Trial
2	Alex Wilber	AlexW@BAPpartners.onmicrosoft.com	Flow for Office 365	Office 365 Enterprise E5	11/29/2016 18:48	FALSE
3	Alex Wilber	AlexW@BAPpartners.onmicrosoft.com	Flow Free	Flow Free	8/24/2017 19:45	TRUE
4	Alex Wilber	AlexW@BAPpartners.onmicrosoft.com	PowerApps for Office 365	Office 365 Enterprise E5	11/29/2016 18:48	FALSE
5	Alex Wilber	AlexW@BAPpartners.onmicrosoft.com	Flow Plan 2 Trial	Flow Plan 2 Trial	8/24/2017 19:45	TRUE
6	Alex Wilber	AlexW@BAPpartners.onmicrosoft.com	PowerApps Trial	PowerApps Plan 2 Free Trial	6/19/2017 18:31	TRUE
7	Nestor Wilke	NestorW@BAPpartners.onmicrosoft.com	Flow for Office 365	Office 365 Enterprise E5	11/29/2016 18:57	FALSE
8	Nestor Wilke	NestorW@BAPpartners.onmicrosoft.com	PowerApps for Office 365	Office 365 Enterprise E5	11/29/2016 18:57	FALSE
9	Grady Archie	GradyA@BAPpartners.onmicrosoft.com	Flow for Office 365	Office 365 Enterprise E5	11/29/2016 19:08	FALSE
10	Grady Archie	GradyA@BAPpartners.onmicrosoft.com	PowerApps for Office 365	Office 365 Enterprise E5	11/29/2016 19:08	FALSE
11	Isaiah Langer	IsaiahL@BAPpartners.onmicrosoft.com	Flow for Office 365	Office 365 Enterprise E5	11/29/2016 18:58	FALSE
12	Isaiah Langer	IsaiahL@BAPpartners.onmicrosoft.com	PowerApps for Office 365	Office 365 Enterprise E5	11/29/2016 18:58	FALSE
13	Patti Fernandez	PattiF@BAPpartners.onmicrosoft.com	Flow for Office 365	Office 365 Enterprise E5	11/29/2016 18:50	FALSE

Figure 23 Output when downloading User Licenses

LICENSING AND LICENSE MANAGEMENT

General purpose, full Power Apps and Power Automate capabilities are licensed on a standalone basis. Additionally, limited Power Apps and Power Automate capabilities are included within various Microsoft 365 (formerly Office 365) and Dynamics 365 licenses. High level overview of the licensing structure is provided below.

Per user, per app or Per flow (minimum purchase of 5 flows)	Per User	Seeded Power Apps (Through Microsoft 365 (formerly Office
---	-----------------	--

			365) and Dynamics 365 user licenses)
Power Apps	Allow individual users to run applications ³ for a specific business scenario based on the full capabilities of Power Apps	Equip users to run unlimited applications based on the full capabilities of Power Apps	Customize and extend Microsoft 365 (formerly Office 365) and/or Dynamics 365 applications (respectively)
Power Automate	Implement flows with reserved capacity that serve unlimited users across an organization.	Allow individual users to create unlimited flows based on their unique needs	Automate business processes and workflows for Microsoft 365 (formerly Office 365) (if licensed), Dynamics 365 (if licensed), and Power Apps

The Microsoft Power Apps and Power Automate Licensing Guide will provide you more details <https://go.microsoft.com/fwlink/?LinkId=2085130>.

As an administrator, you are not required to have a standalone Power Apps or Power Automate license to manage environments.

KEY CONSIDERATIONS FOR PER-APP / PER-FLOW CAPACITY MANAGEMENT

1. How much capacity do I need?
 - a. Is a per user or per app license more cost effective?
 - b. What add-on capacity do I need? E.g. Portal Page Views, AI credits
 - c. How much storage capacity do I need? For database, files, logs
2. Review capacity entitlements from Power Platform admin center
3. Purchase capacity and licenses via the Microsoft 365 admin center, Volume Licensing, or via a partner. Self-service purchase for certain SKUs is also available (see next section for details).
4. Allocate add-on capacity to environments
5. Share and enable* apps & flows to use add-on capacity

SELF-SERVICE PURCHASE

Microsoft is enabling self-service purchase for Power Platform products. This capability will not be available to tenants in the US that are government, nonprofit, or education, at this time.

Customers will be able to make a self-service purchase online from the Microsoft Power BI, Power Apps,

³ Power Apps Per User licensee allows running unlimited apps, Per App license allows end user to run 2 custom apps and access 1 custom portal

and Power Automate websites. Customers will first be asked to enter an email address to ensure they are a user in an existing Azure Active Directory (AD) tenant. Then they will be directed to log in by using their Azure AD credentials. After signing in, the customer will be asked to select how many subscriptions they want to purchase and provide credit card payment. When the purchase is complete, they will be able to start using their subscription. The purchaser will also be able to access a limited view of the Microsoft 365 admin center where they can enable other people in their organization to use the product.

Admins are provided with a way to turn off self-service purchasing on a per product basis via the [MSCommerce PowerShell module](#):

Example script to disable Self Service Purchase for Power Automate

```
Import-Module -Name MSCommerce
Connect-MSCommerce #sign-in with your global or billing administrator account
when prompted
$product = Get-MSCommerceProductPolicies -PolicyId AllowSelfServicePurchase |
where {$_.ProductName -match 'Power Automate'}
Update-MSCommerceProductPolicy -PolicyId AllowSelfServicePurchase -ProductId
$product.ProductID -Enabled $false
```

More details on this option can be found in the [self-service purchase FAQ](#).

STORAGE CAPACITY

Common Data Service capacity (database, file, log and add-ons) are pooled across the tenant and shared amongst all environments and workloads. The first subscription Power Apps, Power Automate, or Dynamics 365 Customer Engagement provides a one-time default capacity entitlement for the tenant. For example, a Power Apps per user plan would set the tenant capacity initially as 10GB of CDS database, 20GB of CDS File and 2GB of CDS log capacity. Each additional licensed user provides an additional per user capacity grant that increases the overall tenant available capacity. There are also capacity add-ons available to purchase additional database, file and log capacity.

In order to create new Power Platform environments with or without CDS there must be at least 1GB of CDS database capacity remaining. Capacity is also consumed by normal CDS storage consumption by storing data, files and logs. As an administrator you can monitor your capacity usage in the admin portal – we will explain this in more detail in the Monitoring Storage Capacity & Add-Ons section of this whitepaper.

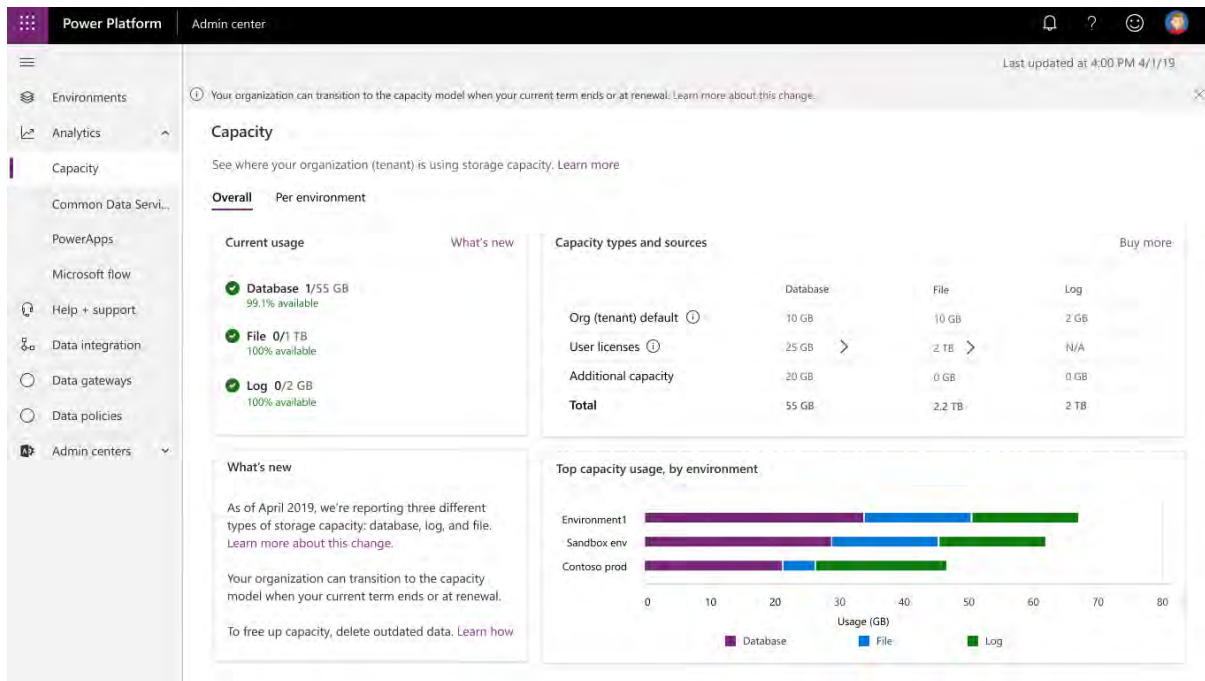


Figure 24 Capacity analytics

API CAPACITY

Another key licensing concept to be aware of is the request limits and allocation. On a daily per user basis, API usage is tracked across Power Apps, Power Automate workflows as well as direct developer API usage. The usage is expected to stay below the API request allocation that is provided based on the type of usage. The allocation as well as usage is tracked at the user level and not at the tenant level. The allocations have been designed so that most users will never exceed the limits. You can monitor basic usage metrics in the admin portal and more detailed usage will be provided in the future. Administrators should work with app makers to help them design their solutions to stay within the limits. If usage for a user continuously exceeds the limit an add-on is available to increase an individual users limit. For more details on request limits and allocations review here <http://aka.ms/platformlimits>.

[Power Apps and Power Automate API request capacity add-on](#) allows customers to purchase additional requests which can be assigned to any user who has a Power Apps/Power Automate license as well as Dynamics 365 license. These can be assigned to an application, and administrative and non-interactive users.

Each capacity add-on provides an additional 10,000 requests/24 hours which can be assigned to any user. Multiple capacity add-ons can also be assigned to the same user.

USE OF CONNECTORS

Canvas apps and flows use connectors to interact with services. Connectors can be standard, premium or custom. To use premium connectors users must be licensed with Power Apps per app or per plan licenses or through a seeded Dynamics 365 license. Users who get Power Apps use rights included with Microsoft 365 (formerly Office 365) licenses can only use standard connectors.

TRIAL PLANS

Trial plans are available for both Power Apps and Power Automate. Free trials last 30 days for Power Apps and 90 days for Power Automate. Users can self-service sign up for these trials in your organization by explicitly visiting the pricing pages or by being prompted when they attempt an action in the apps that require additional licensing.

For Power Automate, an unlicensed user that signs in to flow.microsoft.com will be setup with the free plan. If later they try to perform an action like sharing a workflow, they will be prompted to sign up for a trial. In this example, if the user accepted the offer for trial, they would be signed up for a Power Automate per user trial. This trial would not show up under the user licenses in the Microsoft 365 (formerly Office 365) Portal, however, you would be able to see it in the Power Apps license report discussed in the **What users are already licensed?** section of this whitepaper.

For Power Apps, if a user signs up for a trial, they will be assigned Power Apps user plan trial.

As the administrator, you will likely be assisting users that had started in a trial and either want to continue experimenting or are ready to get a regular license to keep working with the app they are building. If you are moving to a regular license for a user, it would also be a good time to work with them to see if their app should stay where it was built or should be moved according to the environment strategy you adopt. For those not ready to get a full license but want to keep experimenting you could help them get setup on the community plan and help them move their application assets into their new developer environment. It's important to remember trial environments only last 30 days and at the end all resources are deleted if they are not converted to production environments.

POWER APPS COMMUNITY PLAN

In addition to the trial plans, there is also a free Power Apps community plan. This is a special plan that allows individual self-service sign-up and it provides an individual environment that the user can use to build apps and workflows. As the environments are for individual use, there is no ability to share with other users. However, solutions can be exported from this environment and into another environment – it is not recommended to use this type of license for enterprise development purposes.

These environments will show up on the administrator's list of environments and will list the type of environment as "Developer".

Users in your organization can self-service signup for this plan even if they have Power Apps and Power Automate license entitlements via another licensing plan. Signup for the community plan can be found here <https://Power Apps.microsoft.com/communityplan/> and more details on its features here <https://docs.microsoft.com/Power Apps/maker/dev-community-plan>.

Provisioning developer environments can be restricted by raising a support ticket.

LAYERS OF SECURITY

What sets the Power Platform apart from other low-code options that are in use already in your organization already (through Excel or Access) or other Shadow IT, point-solution SaaS providers is that everything is governed and authenticated through Azure Active Directory – you need to sign in with your Work or School Azure AD Account in order to use this service. This means that as an admin, you have full visibility of everything your makers and users do - it is governable, automatable, auditable and manageable by default.

In this section of the paper we are going to look at how the Power Platform handles security from user authentication to authorization which allows users to perform actions with data and services. Conceptually, security in the platform is there to ensure users can do the work they need to do with the least amount of friction, while still protecting the data and services.

The following is a high-level look at how the multiple layers of security make up the security model of the Power Platform:

- Users are authenticated by Azure Active Directory (AAD), and use can be restricted using [conditional access policies](#).
- Licensing is the first control-gate to allowing access to Power Apps components.
- Ability to create applications and workflow is controlled by security roles in the context of environments.
- A user's ability to see and use Power Apps resources is controlled by sharing the application with the user. Sharing of Power Apps canvas apps is done directly with the user or AAD group. Sharing of Power Apps model-driven apps is done via assigning the user the appropriate CDS security role.
- Environments act as security boundaries allowing different security needs to be implemented in each environment.
- Power Automate flows and canvas apps use connectors. The specific connections credentials and associated service entitlements determine permissions when apps use the connectors.
- Environments with a Common Data Service (CDS) instance add support for more advanced security models that are specific to controlling access to data and services in that CDS instance.
- Connector use can be further restricted with Data Loss Prevention (DLP) policies. Cross-tenant inbound and outbound restrictions can also be applied to the connectors

It's important to note, that when accessing data sources via connectors all the underlying security that the data source offers is in addition to the layers of security described above. Power Apps and Power Automate do not provide users with access to the connector data source they don't already have. Users should only have access to data that they really require access to.

AZURE AD CONDITIONAL ACCESS

[Conditional Access policies](#) at their simplest are if-then statements, if a user wants to access a resource, then they must complete an action. Example: A payroll manager wants to access the payroll application and is required to perform multi-factor authentication to access it.

Administrators are faced with two primary goals:

- Empower users to be productive wherever and whenever
- Protect the organization's assets

By using Conditional Access policies, you can apply the right access controls when needed to keep your organization secure and stay out of your user's way when not needed. Conditional Access policies are enforced after the first-factor authentication has been completed.

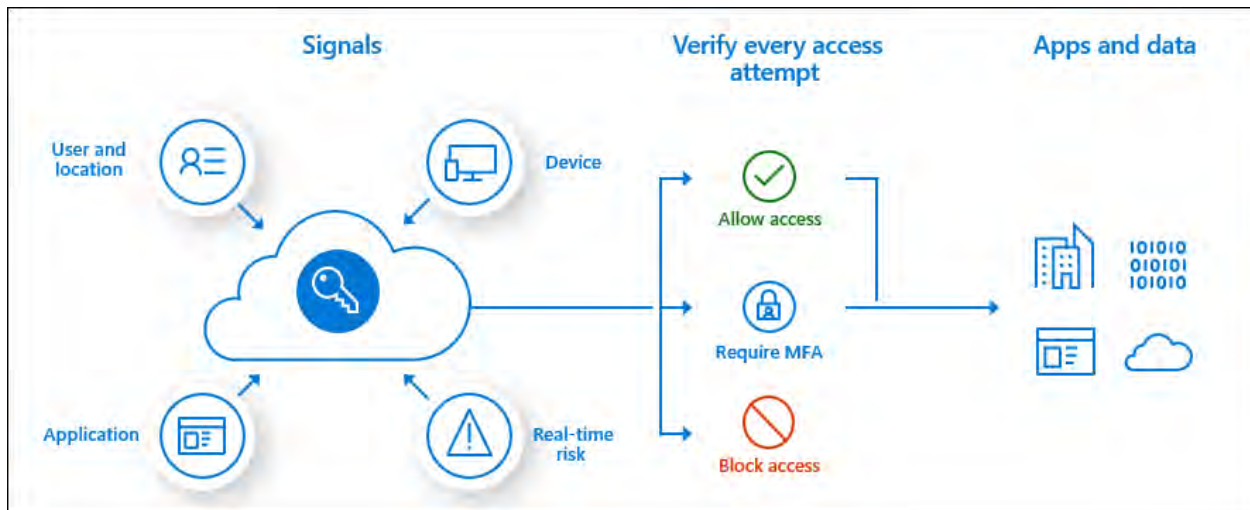


Figure 25 Conceptual Conditional Access process flow

ENVIRONMENT SECURITY ROLES

See **Environment Security Roles**

RESOURCE PERMISSIONS FOR APPS, FLOWS AND CONNECTORS

Power Apps and Power Automate relies on Azure Active Directory (AAD) for authentication. This means that you can leverage the full functionality of AAD to manage and restrict access to users. This includes using conditional access policies and other premium features of AAD. Developers can also register

applications with AAD and use the oAuth2 authorization framework to allow their code to access the platform APIs.

SHARING WITH USERS IN YOUR TENANT

Power Apps Canvas apps can be shared with **individual users**, with **security groups** or with **Everyone in the organization**. As Power Apps does not escalate privileges, you must also manage permissions for the data source or sources on which the app is based, such as Common Data Service or SharePoint. You might also need to share other resources on which the app depends, such as flows, gateways, or connections.

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 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.

By selecting Everyone under share options, the app will be shared with everyone in your organization.

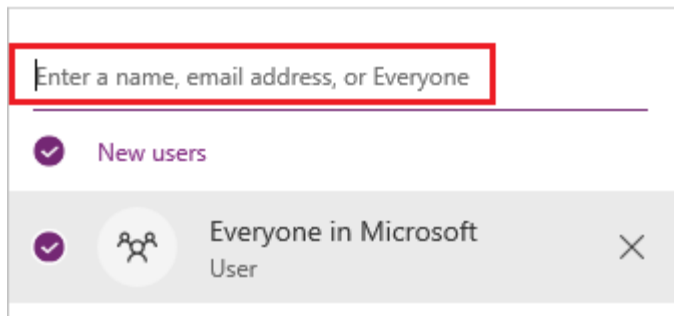


Figure 26 Share Canvas App with Everyone

Sharing with distribution groups is currently not supported.

More detail on sharing Canvas apps is available here

<https://docs.microsoft.com/powerapps/maker/canvas-apps/share-app>

Power Apps model driven apps use role-based security for sharing. The fundamental concept in role-based security is that a security role contains privileges that define a set of actions that can be performed within the app. All app users must be assigned to one or more predefined or custom roles.

You can find more information on role based security in the **Common Data Service Security roles** section of this whitepaper, as well as here <https://docs.microsoft.com/powerapps/maker/model-driven-apps/share-model-driven-app>

Power Automate flows can have individual users, Azure AD Security Groups, Microsoft 365 (formerly Office 365) Groups and SharePoint lists as co-owners. All owners of a team workflow can view the history, manage properties on the workflow, edit the workflow, add and remove other owners (but not the creator), and delete the workflow. Button flows and “For a selected item” SharePoint flows can also be shared with run-only users, which means those users have access to run the flow but not to modify it.

Custom Connectors can also be shared, so that others in your organization can start consuming them <https://docs.microsoft.com/connectors/custom-connectors/share>

SHARING WITH EXTERNAL USERS

Power Apps apps can be shared with guest users of an Azure Active Directory tenant. This enables inviting external business partners, contractors, and third parties to run your company’s apps.

In order to invite a guest user to use Power Apps, [B2B external collaboration](#) for the tenant has to be enabled in Azure Active Directory and you must have a [Guest User](#) created in your Azure AD (Only Admins and users with the Guest Inviter role can add guests to a tenant). The guest user must have a license with Power Apps use rights that matches the capability of the app assigned through either the tenant hosting the app being shared or through the home tenant of the guest user.

As with non-guests, the underlying data source(s) accessed by the app must also be made accessible to the guest.

A detailed break-down of Power Apps canvas app guest access and connectors that support guest access is available on <https://docs.microsoft.com/powerapps/maker/canvas-apps/share-app#share-with-guests>

Guest users can also work with **Power Apps model-driven apps**. You can find more details and the limitations here <https://docs.microsoft.com/dynamics365/customer-engagement/admin/invite-users-azure-active-directory-b2b-collaboration>.

Power Automate allows Azure AD guests to be invited and given permissions to participate in approval processes. Other external users beyond the capability of business guests, including Azure B2C is not currently supported except in portals.

Power Apps Portals handle access a little different because a portal user can either be anonymous, or an authenticated user mapped to either a contact record or a system user record. In the portal configuration you define a web role which defines what the user can access. The users are then assigned to a web role to gain access to protected portal content.

Share this portal



Share with internal Users

To share portals with other users for editing and collaboration, follow the steps below:

1. Create a security role

Go to **Security Roles** and create a new security role that includes all the entities used in your portal.

2. Assign users to the security role

Open the **Users** page under Security for your instance and choose the users you want to share with.

Share with External Users

To share portals with external other users so they can browse and use this portal, follow the steps below:

1. Add users to portal web roles

Go to **Web Roles** and create new or invite existing users to the portal.

2. Invite users

Go to **Contacts** and create new or invite existing users to the portal

Figure 27 Power Apps Portal sharing

You can read in more detail how to configure portal authentication here

<https://docs.microsoft.com/powerapps/maker/portals/configure/configure-portal-authentication>

COMMON DATA SERVICE SECURITY ROLES

One of the key features of the Common Data Service is its rich security model that can adapt to many business usage scenarios. This security model is only in play when there is a CDS database in the environment. As an administrator you likely will not be building the entire security model yourself; but will often be involved in the process of managing users and making sure they have the proper configuration as well as troubleshooting security access related issues.

ROLE-BASED SECURITY

CDS uses role-based security to group together a collection of privileges. These security roles can be associated directly to users, or they can be associated with CDS teams and business units. Users can then be associated with the team, and therefore all users associated with the team will benefit from the role. A key concept of CDS security to understand is all privilege grants are cumulative with the greatest amount of access prevailing. Simply put, if you gave broad organization level read access to all contact records, you can't go back and hide a single record.

CDS teams can be associated with an Azure Active Directory security group or Office group. When this association is established the members of the CDS team are automatically managed by the system. Upon first use of an application that depends on this security the user is automatically added to the CDS team. Additionally, CDS security roles can be configured to treat the role like it was directly assigned to the user.

This allows the user to gain user level privileges even though they acquired them via their association with a CDS team.

To make this easier to configure, when you share a canvas app with an Azure AD group you will be able to select the CDS security roles necessary to use the app. The system will then automatically create a CDS team for you and associate it with the Azure AD security group. The new team will also be automatically associated with the CDS security roles you chose. This simplifies the admin experience and makes this a good way to manage user security while minimizing manual effort to configure.

Security is a complex topic and is best accomplished as a joint effort between the application makers and the team administering the user's permissions. Any major changes should be coordinated well in advance of deploying the changes into the environment.

The **Appendix to CDS security roles** section of this whitepaper covers this in more detail.

CROSS-TENANT INBOUND AND OUTBOUND RESTRICTIONS

With [tenant restrictions](#), organizations can control access to SaaS cloud applications, based on the Azure AD tenant the applications use for single sign-on. For example, you may want to allow access to your organization's Microsoft 365 (formerly Office 365) applications, while preventing access to other organizations' instances of these same applications.

With tenant restrictions, organizations can specify the list of tenants that their users are permitted to access. Azure AD then only grants access to these permitted tenants.

Restricting **outbound** cross-tenant connections can be done using Tenant Restrictions (<https://docs.microsoft.com/azure/active-directory/manage-apps/tenant-restrictions>) that apply to all Azure AD Cloud SaaS apps, or at the API Hub level which would block outbound connections just for Canvas Apps and Power Automate flows. The API Hub level restrictions would currently require a support ticket.

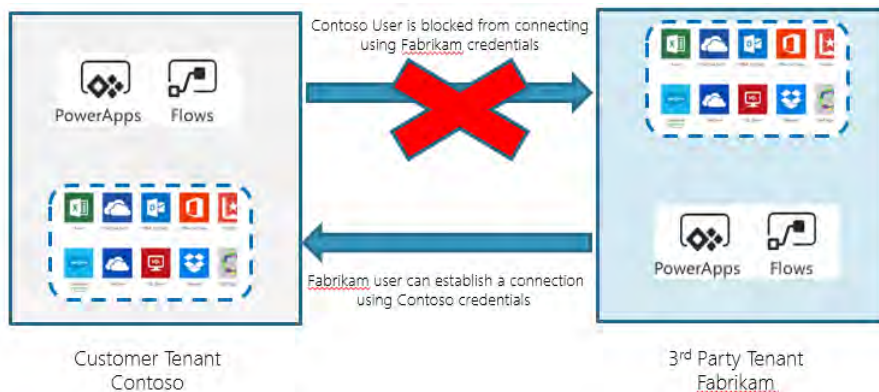


Figure 28 Restrict cross-tenant outbound connections

Restricting **inbound** cross-tenant connections requires a support ticket – this restriction then only applies to Power Apps and Power Automate

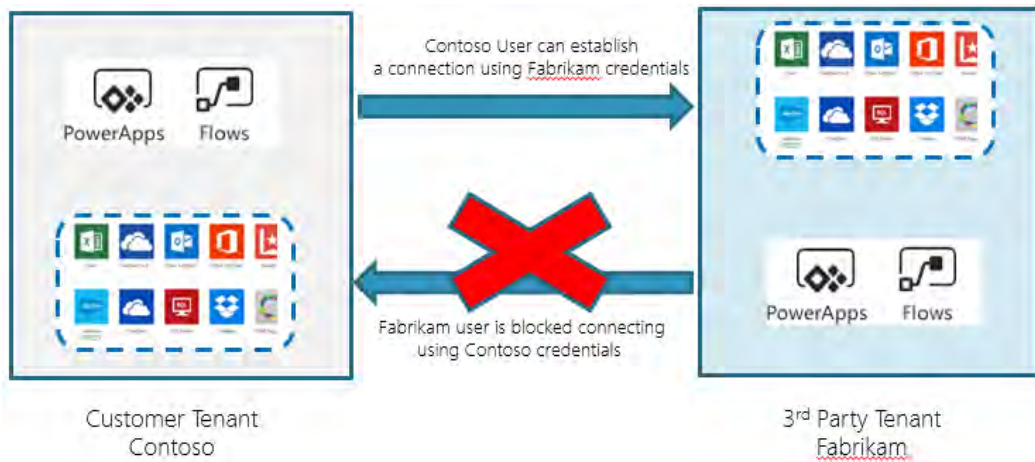


Figure 29 Restrict cross-tenant inbound connections

DATA LOSS PREVENTION POLICIES

Data Loss Prevention policies enforce rules for which connectors can be used together and which are blocked and not usable at all. Connectors are classified as Business Data only, No Business Data allowed or blocked. A connector in the business data only group can only be used with other connectors from that group in the same app or flow. A connector that is blocked can't be used by any app or flow.

- Data loss prevention (DLP) policies act as guardrails to help prevent users from unintentionally exposing the data.
- DLP policies can be scoped at the environment and tenant level offering flexibility to craft policies that are sensible and do not block high productivity.
- Environment DLP policies cannot override tenant wide DLP policies.
- If multiple policies are configured for one environment, the most restrictive policy applies to the combination of connectors.
- By default, there are no DLP Policies implemented in the tenant.
- Policies can't be applied at the user level, only at the environment or tenant level.

- DLP policies are connector aware, but do not control the connections that are made using the connector – in other words, DLP policies are not aware if you use the connector to connect to a development, test or production environment.
- PowerShell and admin connectors can manage policies
- Users of resources in environments can view policies that apply

CREATING NEW DLP POLICIES

Navigate to the **Power Platform Admin Center** <https://aka.ms/ppac> → Data Policies to create and manage your DLP policies.

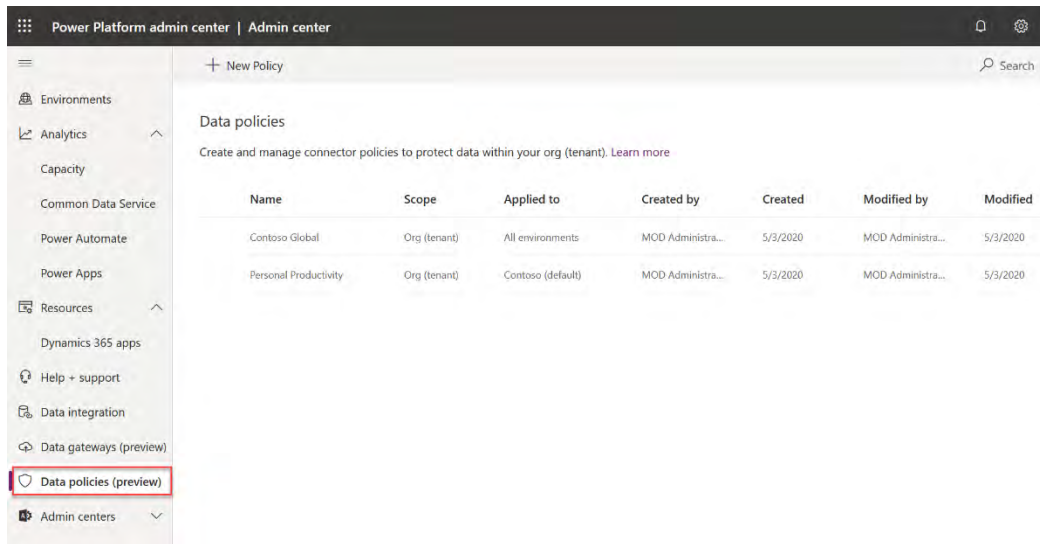


Figure 30 DLP Policies

When you create a new DLP policy you decide on the scope. If you are only an environment administrator, you will see your environments to associate with the DLP policy. If you are a tenant administrator you will have the ability to apply to All Environments, Selected Environments or All Environments EXCEPT.

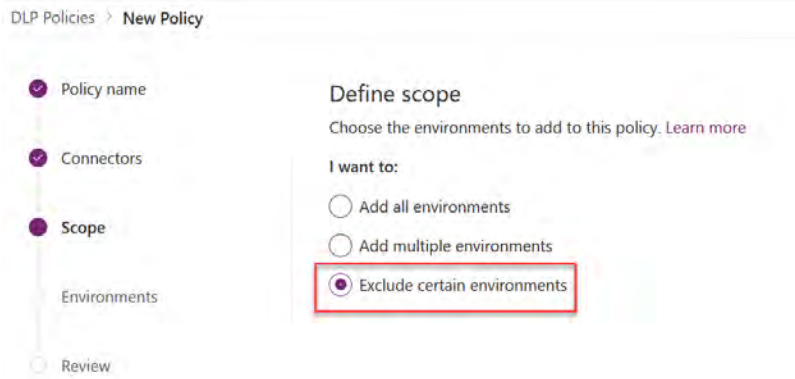


Figure 31 DLP - Environment options

Environment only admins do have the ability to view policies created by tenant admins to understand what might apply to their environment.

CONFIGURING CONNECTORS FOR A DLP POLICY

By default, all connectors are considered part of the no business data allowed list (non-business) and no connectors are included in the business data only group. This effectively means that all connectors can be used with other connectors.

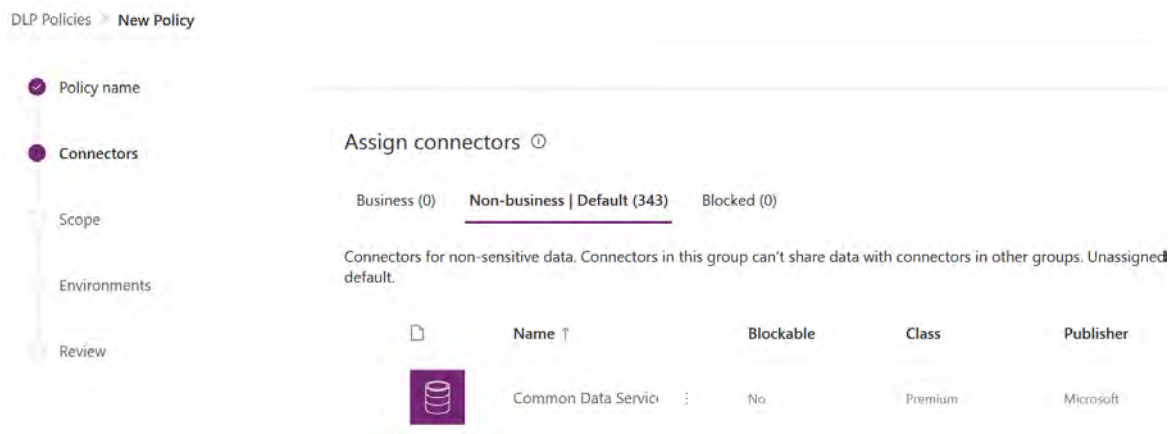


Figure 32 DLP - Configure Data Groups

When new connectors are added they are added to the default category which is no business data allowed. We do not recommend changing the default policy, but having a process in place to analyze new connectors and add them to the appropriate category – for example, a [Power Automate flow could inform you of new connectors](#) prompting you to categorize them.

The blocked group allows you to all together completely block a connector. For example, if you place the Twitter connector in the blocked group, nobody would be able to create a Power App or a Power Automate flow that uses that connector. All third-party connectors can be blocked, and all Microsoft owned premium connectors (except Common Data Service) can be blocked. You can find a list of connectors that can't be blocked [here](#). If a connector is grouped differently by multiple policies, the most restrictive policy is applied. So, if one policy blocks a connector in an environment it is blocked regardless if other policies allow its use.

Using [PowerShell Admin cmdlets](#) or [Power Platform for Admins management connectors](#), you can further automate the management of DLP policies.

Let's look at an example of a tenant-wide DLP policy that had just the Common Data Service connector added to the business data only category, and all other connectors in no business data allowed category.

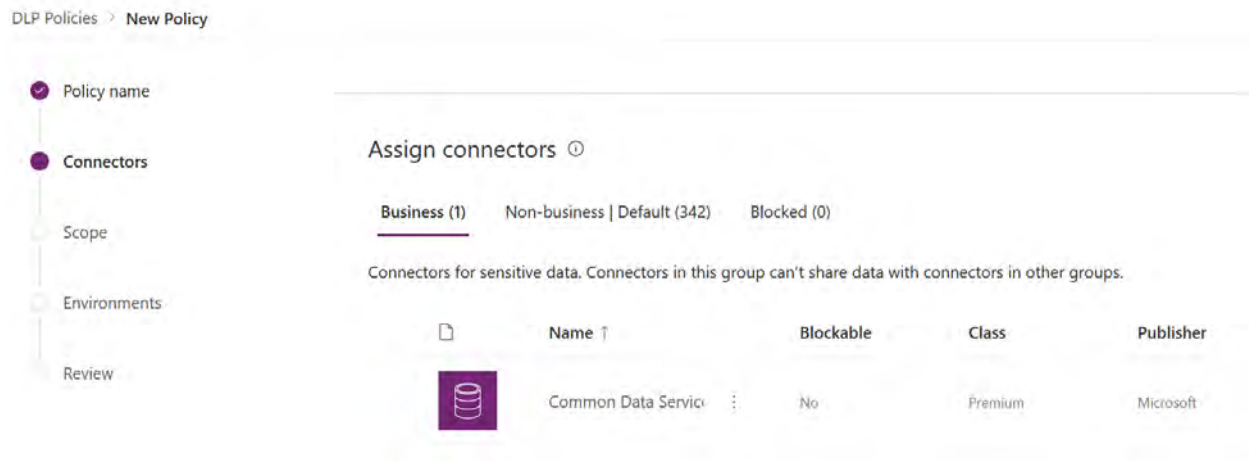


Figure 33 DLP – Sample

Let's look at a few application examples and the outcome of this policy.

Connectors used in application or flow	Impact of DLP
SharePoint and OneDrive	This would be allowed
Common Data Service	This would be allowed
Common Data Service and SharePoint	This would not be allowed
SharePoint and Twitter	This would be allowed
SharePoint, Twitter and Common Data Service	This would not be allowed

IMPACT OF A CHANGE IN DLP POLICY ON EXISTING APPS AND FLOWS

New	Existing
-----	----------

Power Apps	User trying to create a new Power App that violate DLP policies will not be allowed to do so.	Power Apps do not enforce new DLP policies after the app has been created and published. The Power Apps app won't check for DLP policy violations until the maker edits the canvas app again and removes one of the connections and attempts to re-add it since DLP policies only restricts users from adding new connections.
Power Automate	User will not be allowed to create a new Flow that violates a DLP policy	When a flow executes the trigger the Power Automate runtime checks to see if the flow is compliant with all existing DLP policies. If it violates any DLP policy then the Flow will be disabled.

Users creating or editing a resource impacted by the DLP policy will see a message informing of the DLP policy conflict.

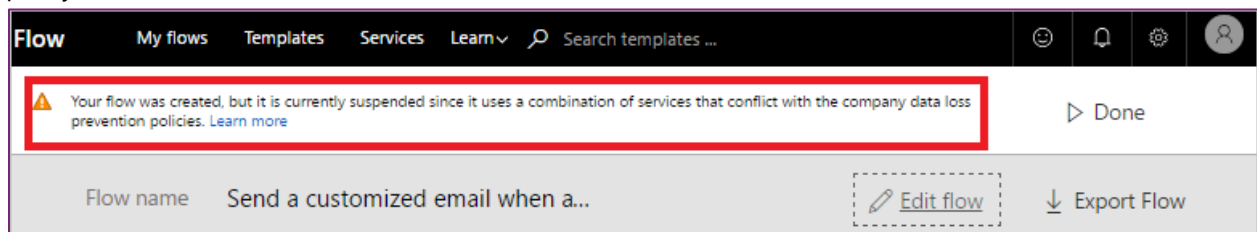


Figure 34 Power Automate flow violating DLP policy

As an administrator you should have a process and plan in place to handle these types of support needs if you are using DLP policies.

Using the DLP Editor in the [Center of Excellence starter kit](#), you can see the impact a change of DLP policies would have on existing apps and can mitigate the risk by reaching out to the maker.

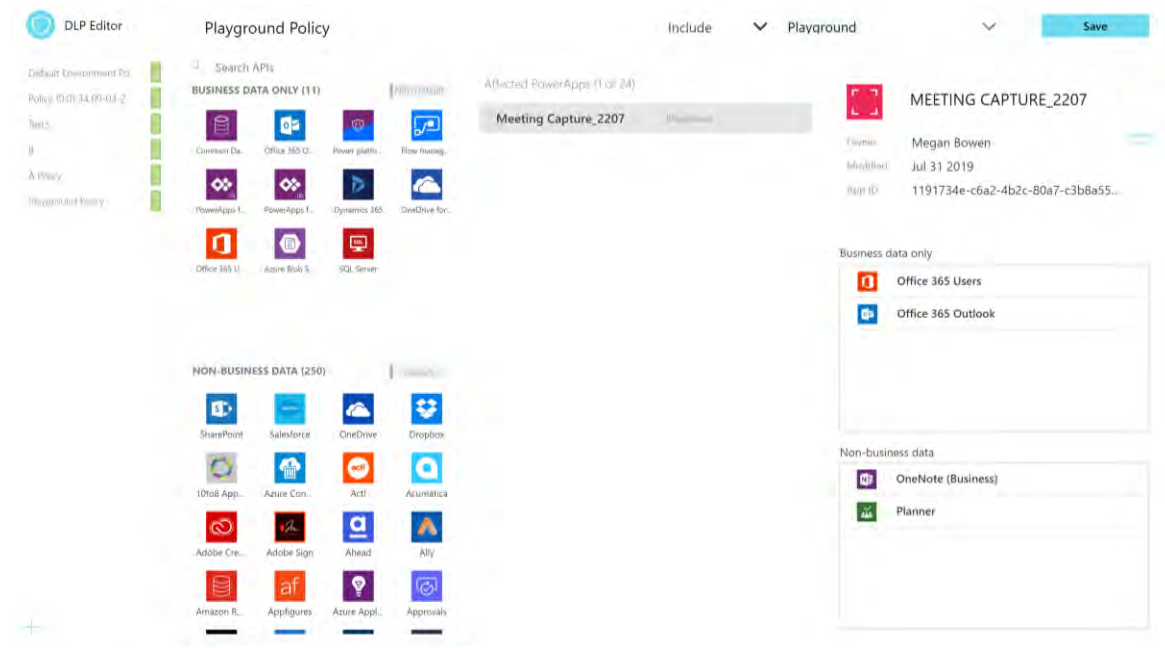


Figure 35 DLP Editor (part of the CoE Starter Kit)

CONNECTOR

By default, custom connectors are not part of the standard configuration capabilities of DLP policies. Unlike the public connectors, custom connectors are scoped to individual environments. Custom connectors must be added to each environment. Once a custom connector is configured in an environment you can use them as part of your DLP policies just like the public connectors. Since custom connectors aren't available to all environments you can't create tenant wide policy's that specify a custom connector.

The [Center of Excellence starter kit](#) has an app that allows users to update policies for these connectors as well, this provides a UI front-end to the PowerShell scripts.

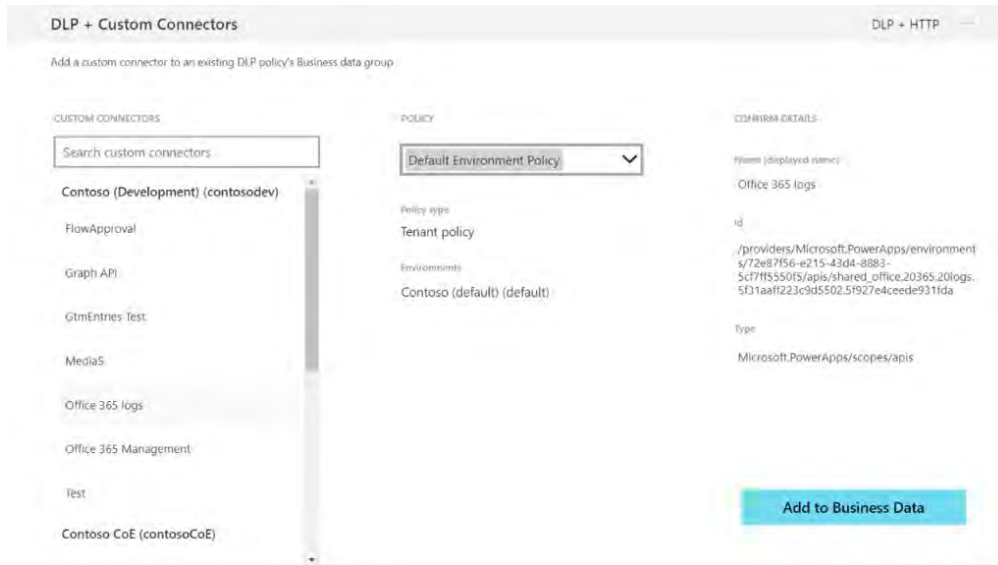


Figure 36 DLP Customizer (part of CoE Starter Kit)

STRATEGIES FOR CREATING DLP POLICIES

As an administrator taking over an environment or starting to support use of Power Apps and Power Automate DLP policies should be one of the first things you set up. This ensures a base set of policies is in place, and you can then focus on handling exceptions and creating targeted DLP policies that implement these exceptions once approved.

We recommend the following starting point for DLP Policies:

1. Create a policy spanning all environments that blocks all unsupported non-Microsoft connectors and classifies all Microsoft connectors as 'Business Data'⁴
2. Create a policy for the default environment (and other training environments) that further restricts which Microsoft connectors are classified as 'Business Data'
3. Create additional policies or exclude those environments from policies #1 and #2 above that permit certain connectors or connector combinations to be used for specific environments

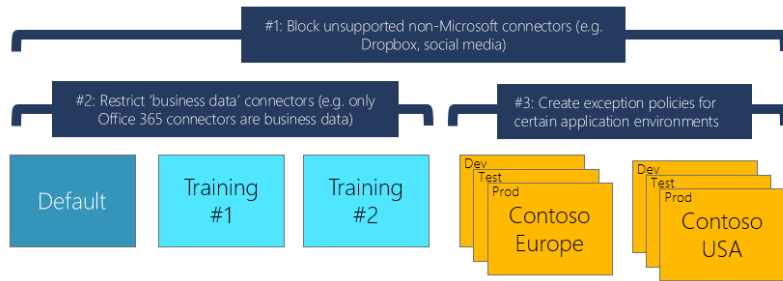


Figure 37 Starting point for DLP policies

With this in place, plan how to handle exceptions – you can:

1. Deny the request.
2. Add the connector to the default DLP policy.
3. Add the users' environments to the All Except list for the global default DLP and create a user specific DLP policy with the exception included.

The ability to block third-party (non-Microsoft) connectors is coming for April 2020. The capability will allow admins to block all third-party non-Microsoft connectors (non office365 and Azure, SQL connectors) at the tenant level or environment level.

- Admin will be able to block third-party connectors all at once or in groups or individually
- Once a blocked policy has been created and made the default category, all new third-party connectors will be blocked automatically.
- Admin can edit this policy and move a third-party connector from blocked to the "Business Data Allowed" group.
- In addition, all custom and http connectors can also be blocked using this PowerShell cmdlet.

CALL TO ACTION

As an administrator you should

- Establish an environment strategy, and a process for requesting access or creation of environments
- Set up data loss prevention policies early, and come up with a plan on how to manage exceptions
- Regularly check for new connectors and move them into the appropriate (Business Data only or No Business Data Allowed) category of your DLP Policies
- Restrict the creation of net-new trial and production environments to admins
- Rename the Default environment to "Personal Productivity"

MONITOR

In this section we will focus on typical tasks that you will be responsible for when administering the Power Platform. We've seen many use-cases being requested during our Governance briefings or running Admin-in-a-day trainings with our customers. Nevertheless, in terms overall possible scenarios around monitoring, alert and actions in a day-to-day manner, we reduced the amount of content in this whitepaper to address:

- Review out of the box monitoring capabilities
- Check service and environment health
- Alert on security permissions or compliancy regulations
- Performing typical actions to ensure security, healthiness and a safe citizen developer environment

We encourage you to visit our Admin-in-a-day content, that can be found on GitHub repository (<https://aka.ms/powerapps/admininaday>) and walk you through some many more examples via hands-on-labs with step-by-step instructions. It is important to understand, that each company has their own operational model and requirements around a citizen app development platform, therefore fulfilling those with the help of using the platform capabilities itself in terms of custom apps or flows can be seen as a best practice. Out of box tooling around monitoring, alert and actions falls into the following three categories:

The Admin portal (aka.ms/ppac) offers an interactive experience for performing administrative tasks. This is typically considered the primary path for completing administrative activities. From a monitoring point of view this channel is used mostly for ad hoc interactive discovery. Additionally, some admin tasks would need to have access to Microsoft 365 (formerly Office 365) Admin Center (<https://admin.microsoft.com/>)

PowerShell cmdlets offer a way to automate both management and monitoring tasks using PowerShell. These cmdlets can be used in a sequence to automate multi-step administrative actions. Note, that from roadmap perspective PowerShell cmdlets will be available first, before enabling administration capabilities via UI or via the Management and Admin Connectors. Check-out (<https://www.powershellgallery.com/>) to get the latest package.

Management and Admin Connectors offer the ability to use the platform's own tools to manage and monitor itself. Part of the out of the box available 300+ connectors and approval process capabilities are five admin specific connectors, we'd like you to familiarize with.

- The **Power Automate Management connector** is specifically designed to help with administrative management and monitoring (<https://docs.microsoft.com/connectors/flowmanagement/>).

- **Power Automate for Admins** allows you to perform typical admin actions, such as disabling a flow or deleting a flow (<https://docs.microsoft.com/connectors/microsoftflowforadmins/>)
- **Power Apps for Admins** connector to set permissions on Power Apps or set permissions to a certain connector being used by this app (<https://docs.microsoft.com/connectors/powerappsforadmins/>)
- **Power Apps for App Makers** which also can be used by the makers themselves, though some actions being an overlay to administrative tasks, such as settings permissions to a Power Apps app as mentioned previously (<https://docs.microsoft.com/connectors/powerappsforappmakers/>)
- **Power Platform for Admins** to perform tasks against platform components, such as creating an environment or provisioning a CDS database or creating a DLP policy for a specific environment (<https://docs.microsoft.com/connectors/powerplatformforadmins/>)

The **Center of Excellence starter kit** offers a template implementation using the management and admin connectors and comes with a Power BI dashboard that can be leveraged to gain tenant wide insights.

WORKING WITH THE ADMIN PORTALS

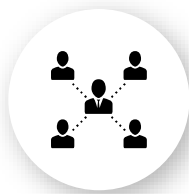
In a perfect world as an administrator you would only visit a single portal to perform all your administrative tasks. But given the scope and breadth of the different products involved and their differing release cycles, there are multiple portals with which you will interact. The following outlines the different portals and the most common tasks you perform there.

Portal	Common Tasks
Power platform admin center https://admin.powerplatform.microsoft.com	The new unified administrative portal for Power platform admins. You can use the shortcut aka.ms/ppac . Some of the tasks you can perform here are: <ol style="list-style-type: none"> 1. Environment Management 2. Manage Support Requests 3. Monitor Analytics 4. Manage Data Gateways 5. Manage Data Policies
Microsoft 365 admin center https://admin.microsoft.com	Here you will manage users and their license assignment as well as you can launch into many of the individual admin centers from here.
Microsoft Azure https://portal.azure.com	Advanced Azure AD management tasks like conditional access is managed here. Also, if you support any developer application registration it is also done here. This is also where you start setup of your on-premises gateway.
Security & Compliance Center https://protection.office.com	In addition to the general compliance tasks, administrators can come here to search the Audit log to see Flow audit events

Partners helping their customers manage their cloud services can use delegated administration capabilities to access the admin portals. You could also do this by having a user in the customer's tenant and have either a service admin level account or have been assigned the equivalent of an environment administrator.

OUT OF BOX ANALYTICS IN POWER PLATFORM ADMIN PORTAL

From the Power Platform admin portal (aka.ms/ppac), you can view tenant level analytics to help you manage capacity and troubleshoot problems. The analytics section on the portal allows you to drill down into Capacity, Common Data Service, Power Automate and Power Apps analytics. Any service admin or environment admin has access to the analytics. The data is refreshed daily or more frequently in some cases and is retained for the prior 28 days. The analytics section follows three design principles, we wanted administration to be equipped with:



Adoption



Usage



Health

Following these principals, you will be enabled to ensure high quality adoption, usage and overall health of the citizen application development platform in your organization.

MONITORING STORAGE CAPACITY & ADD-ONS

The capacity section of analytics allows you to monitor storage capacity use and availability in your tenant. From the all up view across all the environments you can drill down into the individual environments for details such as top entity using storage on a timeline view. Sign into the Admin center and select **Analytics** > **Capacity** in the left-side navigation pane.

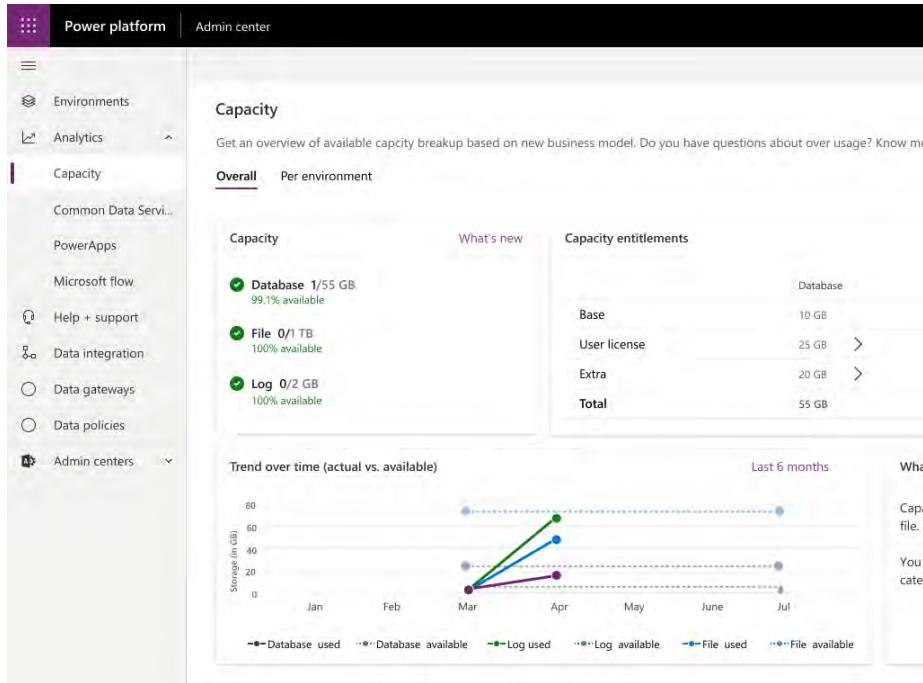


Figure 38 Screenshot Capacity section

If your organization has purchased capacity add-ons, an **Add-ons** tile appears on the **Capacity** screen in the [Power Platform Admin center](#).

The **Add-ons** tile shows summary information about the capacity add-ons that your organization has.

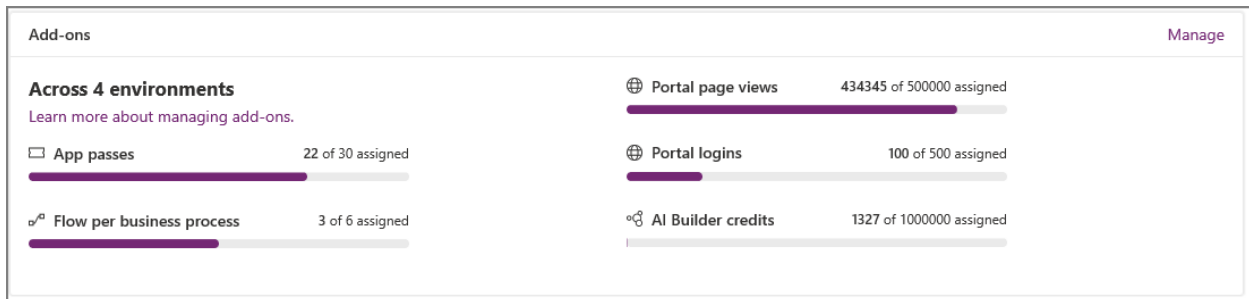


Figure 39 Add-on capacity

To allocate capacity to an environment, select **Analytics > Capacity** and then go to the **Add-ons** tab to select Manage add-ons

Manage add-ons

Assign add-ons to environments so that people can do more with their apps, flows, portals, and AI models. [Learn more](#)

Environment

Production - PowerApps - US

App passes

Give unlicensed users access to PowerApps apps and flows.

0 50 remaining

Flow per business process

Assign capacity so your people can use business process flows.

0 10 remaining

Portal page views

Allow anonymous visitors to use your portals.

0 100000 remaining

Portal logins

Manage the number of logins allowed on your portals.

0 100 remaining

AI Builder credits

Assign AI processing credits to your PowerApps apps and flows.

10 999900 remaining

Save Cancel

Figure 40 Manage add-ons from Power Platform Admin Center

As an admin, you can restrict who can allocate add-on capacity to environments.

1. Sign in to the Power Platform Admin center at <https://admin.powerplatform.microsoft.com>.
2. Select the **Gear** icon (⚙️) in the upper-right corner of the Power Platform site.
3. Under **Who can allocated add-on capacity to environments**, select **Only specific admins**.
If set to Only specific admins, only Microsoft 365 (formerly Office 365) Global Admins, Service Admins and Delegated Admins will be able to allocate add-on capacity.
Note: This option will only show up if add-on capacity is available.

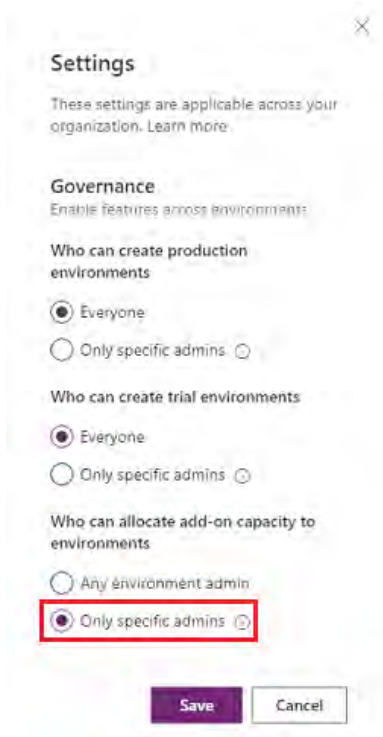


Figure 41 Restrict who can allocate add-on capacity

CALL TO ACTIONS:

As an administrator you should

- regularly check the capacity available to ensure new environments can be created by users in your tenant.
- review the top storage used by environments to ensure there aren't any environments show up that aren't expected.
- Using the top database, file and log charts for individual environments you can look for any unexpected spikes in usage that is unexpected.
- Since app makers often don't have access to production environments, you can download images or csv files of the data to share with them as appropriate.
- review add-ons capacity such as Power Apps app passes, Flow per business processes, Portal page views, Portal Logins or AI Builder credits and assign those capacity to specific environments

MONITORING COMMON DATA SERVICE

The Common Data Service (CDS) section will provide details on CDS usage in the selected environment if it is provisioned. You can change environment by click on *Change filters* and select your environment as

well as adjust date range, data will be presented to you. The data can only be presented in the prior 28-day timeframe

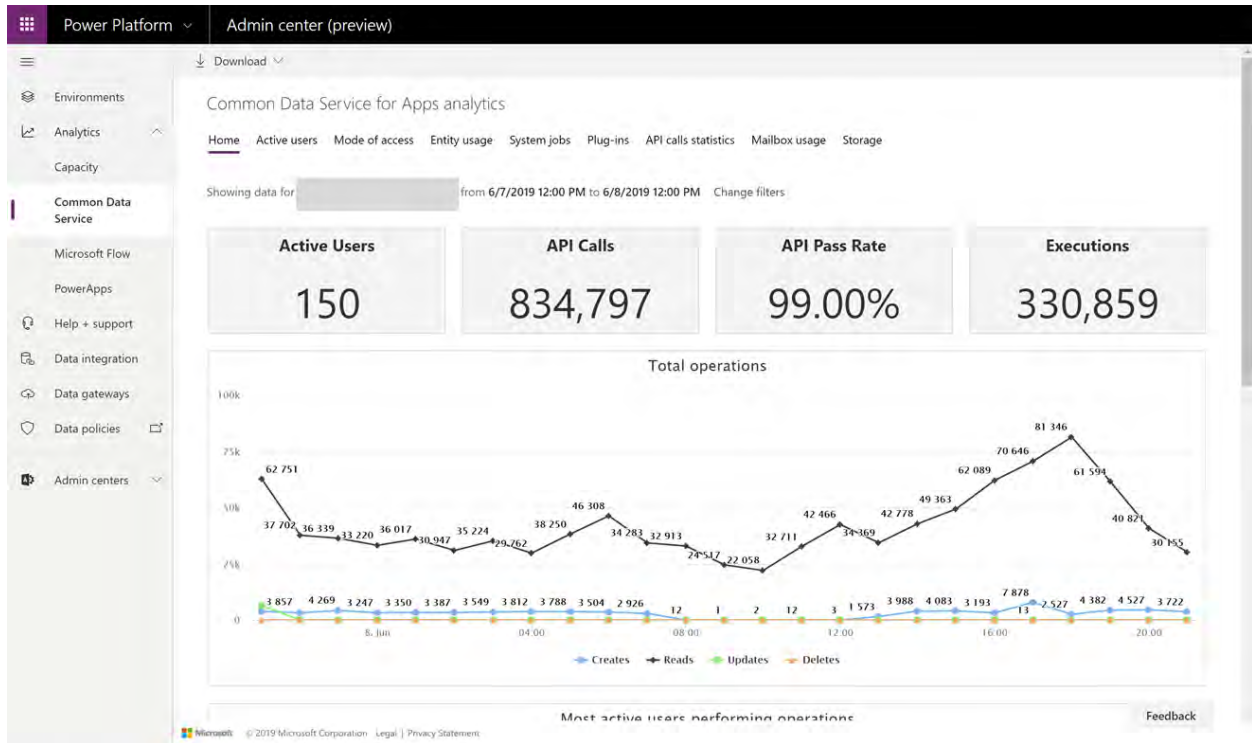
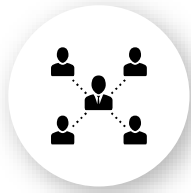


Figure 42 Analytics - Common Data Service

Common Data Service analytics helps you monitor the following:



Adoption

- Number of Active users
- Active user trends
- Top active users
- Mode of access
 - o By OS
 - o Device type
 - o Browser
 - o and more



Usage

- Most used out of the box entities
- Most used custom entities
- Activities performed (CRUD)
- System jobs, Plug-in's and API call usage



Health

- System jobs analysis
 - o Pass rate
 - o Throughput
 - o Top failures
 - o Backlog
- Plug-in analysis
 - o Pass rate
 - o Execution time
 - o Top failures
- API calls analysis
 - o Pass rate
 - o Most used APIs
 - o Top failures

CALL TO ACTIONS:

As an administrator you should

- Setup Azure Conditional Access in line with organization policies to restrict of access modes by OS or device type and browsers
- assist app makers in terms of activities performed on Common Data Service database
- ensure overall healthiness of the platform by regular checks on system jobs operating, Plug-ins being used by app makers and API calls performed against Common Data Service

MONITORING POWER AUTOMATE

This section provides analytics on the Power Automate flows that you have within the environment set in the filter.

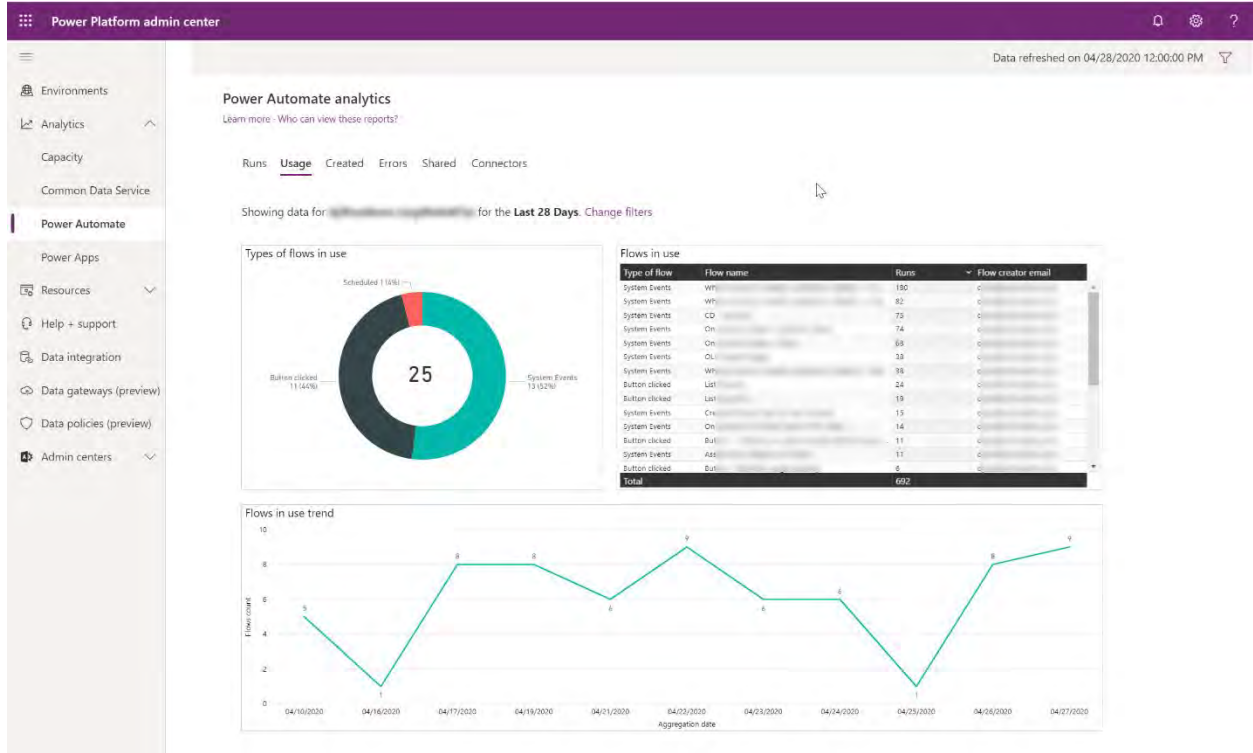
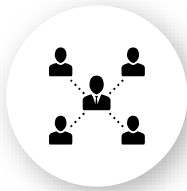


Figure 43 Analytics - Power Automate

Power Automate analytics helps you monitor the following:



Adoption

- Types of flows created
 - o Button clicked
 - o Scheduled
 - o System events
- Flow creation trends
- Created flow list view



Usage

- Run metrics
 - o Active runs, success and failure
- Flows in use
- Flows in use trend
- Shared flows and types, flow names, number of shares
- Connectors by runs, number of connections, flows involved and drill down on flows



Health

- Total number of errors
 - o Split by error type
- Errors across flows
 - o List of errors by flow name
- Power BI capabilities to drill down

CALL TO ACTIONS:

As an administrator you should

- monitor usage and look for insights related to types of flows that are in use
- watch for errors by error type to look for common problems that may exist
- spot data usage that isn't expected by drilling into connector usage and adjust your data loss prevention policies for that environment to ensure an overall health of the platform

MONITORING POWER APPS

The Power Apps section of analytics focuses on canvas apps. The data is presented one environment at a time and you can switch between environments as well as date range of the data – again 28-day time window.

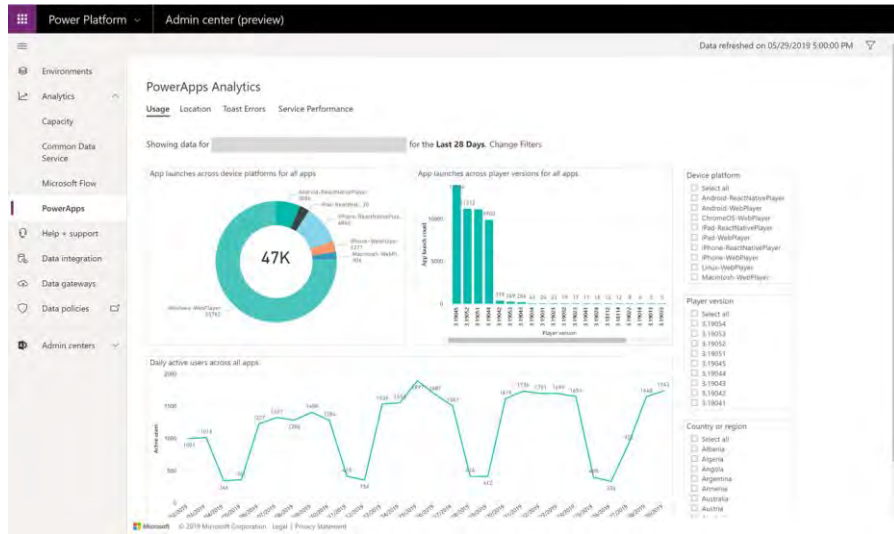
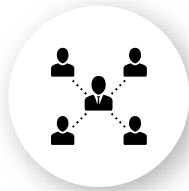


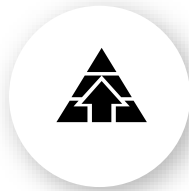
Figure 44 Analytics - Power Apps

Power Apps analytics helps you monitor the following:



Adoption

- Active User
 - o How many users are using the app?
 - o Daily active usage
- Location
 - o Where is the usage of apps?
 - o Drill down by country



Usage

- App usage
 - o App launches across device platform
 - o App launches across player version



Health

- Service Performance
 - o Best performing service
 - o Least performing service
 - o Service response time
 - o And more
- Error reporting
 - o Apps by toast error count
 - o Toast error trend
 - o Error breakdown by http status

CALL TO ACTIONS:

As an administrator you should

- watch overall adoption by monitoring insights into how much the apps are being used and when
- check out app launches in terms of being used in browser or via mobile player version and on which platforms. Following your device strategy, you could ensure users using the latest player edition
- regular monitor overall service performance to ensure user run-time experience when interacting with the platform services

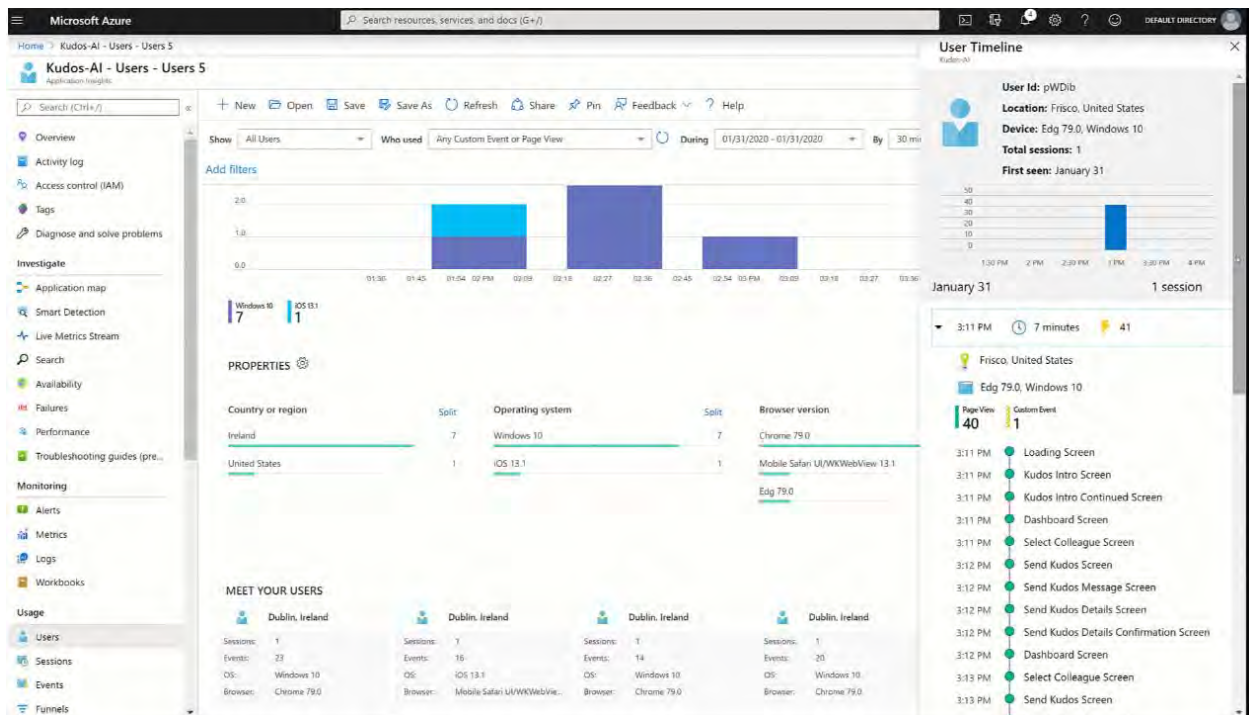
Tip: In addition to the Admin Analytics we looked at in this section, owners of apps and flows can also get maker analytics from make.powerapps.com and flow.microsoft.com. These include usage numbers (total runs and trends) and errors (by day, error type and details) and in some cases troubleshooting help. Makers can also self-sign up for proactive email updates that would include usage information for their top performing apps. The emails also include recommendations on how to make improvements using new platform features. You can access those analytics if the app or flow is shared with you. As an administrator you can always add yourself to the app or flow to gain access.

LOG POWER APPS TELEMETRY USING APPLICATION INSIGHTS

In addition to the analytics already collected when Power Apps are run, makers can connect their Power Apps canvas apps to Azure Monitor Application Insights. Application Insights collects detailed telemetry to help diagnose problems and gain understanding of how the application is being used. This allows detailed analysis of data including:

- Number of users who viewed the app.
- Number of sessions by the users for the app.
- Number of events logged for the app.
- Operating systems and browser version details of the users.
- Region and location of the users.

More advanced analysis can be done using Application Insights features like Cohorts, Impact analysis, Retention analysis and Usage flows. These provide detailed insights on app usage patterns and help you make decisions on how to evolve the app.



App makers can also log custom trace events from within their application logic. This allows capturing application specific information in the telemetry. These trace events can be correlated with other application activity to help troubleshoot and understand how users are interacting with the application.

You can find more details about application insights [here](#).

POWER APPS AND POWER AUTOMATE ACTIVITY LOGGING VIA MICROSOFT 365

Power Apps and Power Automate activities can be tracked and viewed from the Microsoft 365 (formerly Office 365) Security and Compliance Center at <https://protection.office.com>. This allows seeing when activities like apps or flows are created, edited or deleted along with other key activities. These logs can be used manually for discovery and review and can also be accessed via API to automate more complex scenarios. The following are the Power Automate and Power Apps activities that are logged:

Power Automate

Power Apps

- | | |
|--|---|
| <ul style="list-style-type: none"> • Created flow • Edited flow • Deleted flow • Edited permissions • Deleted permissions • Started a paid trial • Renewed a paid trial | <ul style="list-style-type: none"> • Created app • Edited/save app (draft) • Published app • Deleted app • Restored an app from app version • Launched app • Marking app as featured • Marking app as hero • Edited app permissions • Deleted app permissions |
|--|---|

In order to take advantage of the activity logging, you must have an Office E3 or greater license and you must turn on audit logging in the tenant before data is available for viewing.

This is completed via the following PowerShell commands and you can find full details on the steps here <https://docs.microsoft.com/microsoft-365/compliance/turn-audit-log-search-on-or-off>

- Enable-OrganizationCustomization
- Set-AdminAuditLogConfig -UnifiedAuditLogIngestionEnabled \$true

Non service administrators also need permissions to view

- E.g. Add-RoleGroupMember "Compliance Management" -Member user1

If your organization uses a Security Information and Event Management (SIEM) server you can learn how to enable integration with activity logging here <https://docs.microsoft.com/microsoft-365/security/office-365-security/siem-server-integration> .

You may also find Microsoft Compliance Manager helpful to manage your compliance efforts across Microsoft cloud services in a single place. More details about Compliance Manager can be found here <https://aka.ms/compliancemanager> .

You have to be assigned the View-Only Audit Logs or Audit Logs role in Exchange Online to search the Microsoft 365 (formerly Office 365) audit log. By default, these roles are assigned to the Compliance Management and Organization Management role groups on the Permissions page in the Exchange admin center. Note Global administrators in Microsoft 365 (formerly Office 365) are automatically added as members of the Organization Management role group in Exchange Online.

REVIEWING LOG EVENTS

In the compliance center Audit Log Search administrators can now search and view Power Apps and Power Automate events. Using the portal, you can choose what you want to search and a time window.

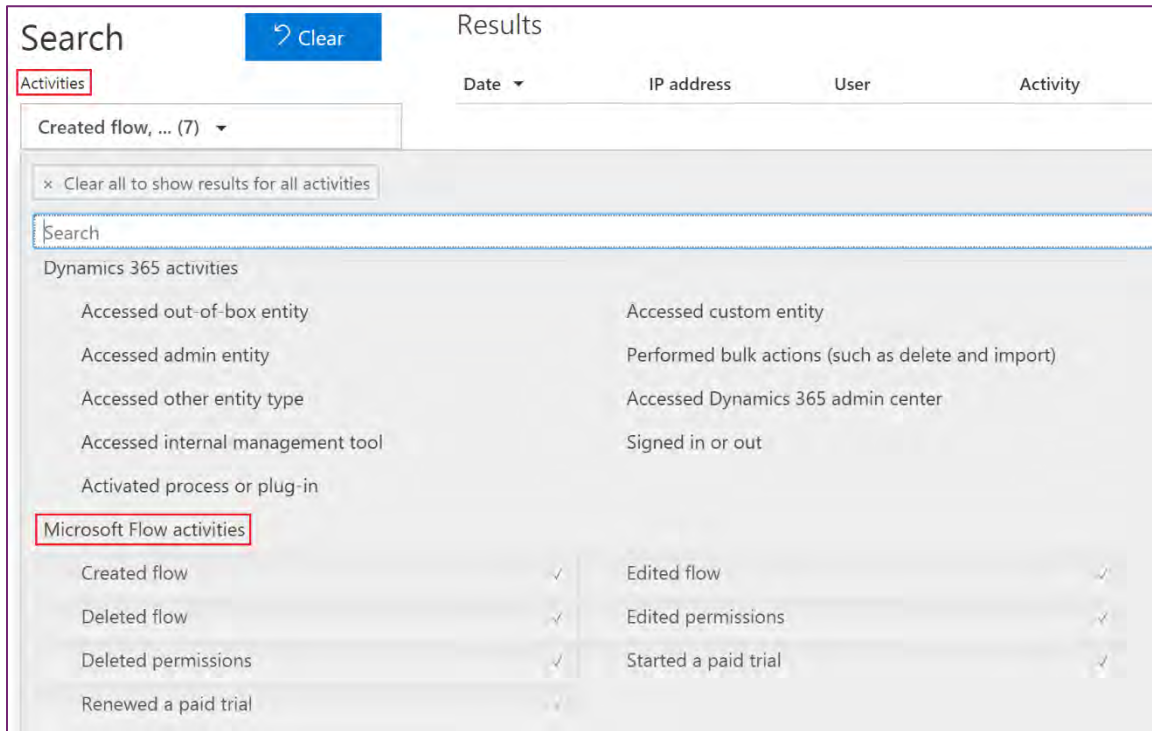


Figure 45 Microsoft 365 Activity Logging

From the query results you can drill down into an item you get a details page with the following type of information.

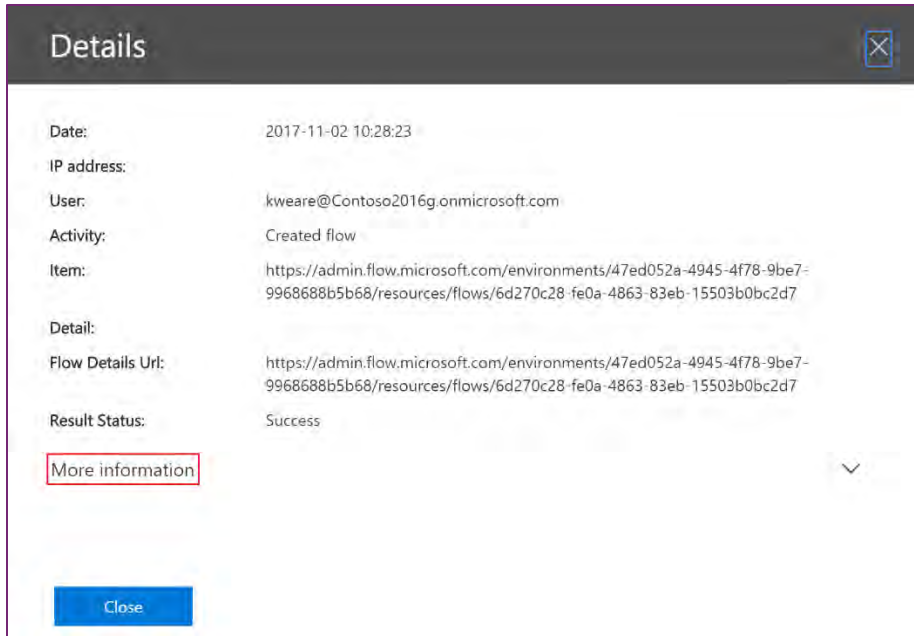


Figure 46 Microsoft 365 Activity Logging Details

Clicking on the More Information drills down into the additional details:

More information	
CreationTime:	2017-11-02T17:28:23
FlowConnectorNames:	Recurrence,Compose,MSN Weather,MSN Weather,lf,Notifications,Notifications,Notifications,Notifications
FlowDetailsUrl:	https://admin.flow.microsoft.com/environments/47ed052a-4945-4f78-9be7-9968688b5b68/resources/flows/6d270c28-fe0a-4863-83eb-15503b0bc2d7
Id:	caa9cb79-c8f9-4c69-96be-f1f390f1d273
LicenseDisplayName:	
Operation:	CreateFlow
OrganizationId:	2f6e98fd-ed85-416f-b7d4-94ad61065d0f
RecipientUPN:	
RecordType:	30
ResultStatus:	Success
SharingPermission:	1
UserId:	kweare@Contoso2016g.onmicrosoft.com
UserKey:	kweare@Contoso2016g.onmicrosoft.com
UserType:	0
UserTypeInitiated:	1
UserUPN:	kweare@Contoso2016g.onmicrosoft.com
Version:	1
Workload:	MicrosoftFlow

Figure 47 Microsoft 365 Activity Logging Details (More)

Audit data is retained for 90 days. You can do exports of the data allowing you to move it into Excel or Power BI for further analysis. You can find a complete walkthrough of using the audit information here <https://flow.microsoft.com/blog/security-and-compliance-center/>

CALL TO ACTIONS:

As an administrator you should

- Carefully think about audit data via Microsoft 365 (formerly Office 365) security & compliance center as in addition to other mechanism for monitoring, you can create new alert policies that can be used for ensuring overall health of your citizen app development platform
- further explore Power Automate capabilities that would help you take more actions out of the email alert and automate additional steps ensuring Power Platform service health

COMMON DATA SERVICE AUDIT LOGGING

When Common Data Service (CDS) is used in addition to activity logging you also have audit logging available for actions in CDS. This includes create, update, delete operations on records as well as changes to CDS metadata. You can read in depth on what can be audited here

<https://docs.microsoft.com/powerapps/developer/common-data-service/auditing-overview> .

For auditing to be captured, it must be enabled in the following three places:

- In the admin portal environment settings via <https://aka.ms/ppac>

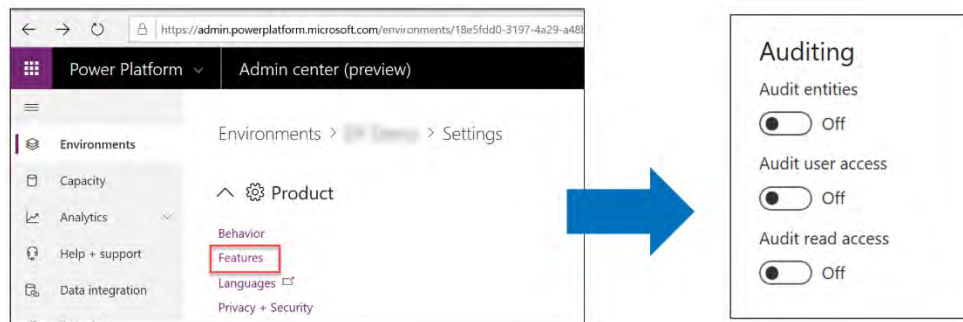


Figure 48 CDS - Enable Auditing

- The entity property must be enabled for auditing

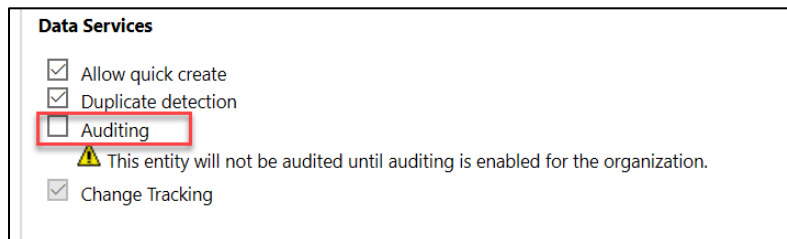


Figure 49 CDS - Entity Auditing

- The field on the entity must be enabled for auditing

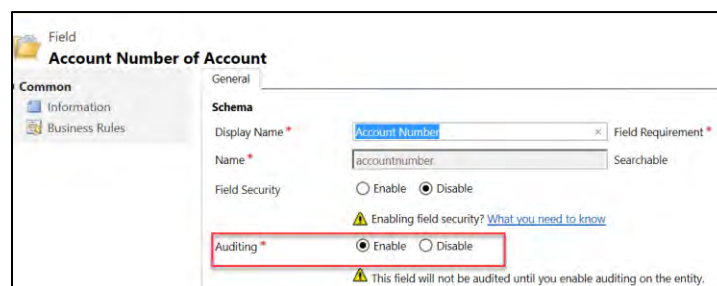


Figure 50 CDS - Field level Auditing

You will need to coordinate with the app makers to ensure the entities and fields are properly configured to support auditing of the data. It is also helpful in some scenarios to turn off auditing on some fields that change frequently and aren't significant to track as it can reduce the volume of audit that data that is captured.

CALL TO ACTIONS:

As an administrator you should

- Know that Audit logging can be helpful in tracking down complex business logic problems that are a result of too many updates or conflicting updates occurring
- Frequently review the logging data as it can provide help in troubleshooting logic problems. So, having some level of audit logging enabled ahead of the need is helpful to expedite problem solving further explore Power Automate capabilities that would help you take more actions out of the email alert and automate additional steps ensuring Power Platform service health

ALERT AND ACT

ALERT & ACTION VIA POWERSHELL OR POWER AUTOMATE LEVERAGING MANAGEMENT CONNECTORS

One of the unique things about administering the Power Platform is you can use the same components to automate administration tasks and to alert the admin team when action is needed. Using the PowerShell cmdlets or the management connectors you can build flows and apps that help you do implement your governance policies. Examples also can be found, when installing and testing the Center of Excellence Starter Kit (aka.ms/CoEStarterKitDownload) or using our Admin-in-a-Day hands on labs which can be found on GitHub (<https://github.com/microsoft/powerapps-tools/tree/master/Administration/AdminInADay>)

AUTOMATION OF TASKS WITH POWERSHELL

The PowerShell cmdlets allow you to do similar tasks that you would do with the admin portals but do them in scripting where you can sequentially execute multiple commands or pipe output from one to automate common tasks. There are multiple PowerShell cmdlets that you can work with. The following is an overview of each that you would likely interact with.

PowerShell cmdlet library	Common Tasks
Power Apps cmdlets https://docs.microsoft.com/PowerApps/administrator/PowerApps-powershell	Designed for app makers and administrators to automate tasks with environments and associated apps, flows and connectors. Note: These cmdlets are currently in preview.
Microsoft 365 (formerly Office 365) 365 cmdlets https://docs.microsoft.com/office365/enterprise/powershell/getting-started-with-office-365-powershell	These are focused on Microsoft 365 (formerly Office 365) related tasks and can be used to automate user-related actions and tasks, for example, assignment of licenses.
Dynamics 365 cmdlets https://docs.microsoft.com/powershell/dynamics365/customer-engagement/overview	These are useful if you have any environments with CDS databases. Modules include support for using the CDS online admin API, as well as to automate solution

Microsoft Azure cmdlets

<https://docs.microsoft.com/powershell/azure/overview>

deployment to the CDS instances.

The Azure cmdlets are useful if you are including any Azure components in your overall solution. This could also be used to script setup of the on-premise application gateway.

It is not uncommon to build PowerShell scripts to do bulk operations on users, environments or their resources that use a combination of all the above cmdlets.

GETTING STARTED WITH POWERSHELL

You can follow our online documentation <https://docs.microsoft.com/power-platform/admin/powerapps-powershell#powerapps-cmdlets-for-administrators-preview> if you're a first-time user of PowerShell cmdlets for Power Platform.

COMMON POWERSHELL TASKS

Displaying a list of environments

```
Get-AdminPowerAppEnvironment
```

This will give you key information such as the Display Name and GUID of the environment. This is often what is needed for follow on operations.

Adding parameters such as -Default will allow you to generically find the default environment in the tenant

```
Get-AdminPowerAppEnvironment -Default
```

Using the GUID you got back (which is the non-display name for the environment) you can drill into details of that specific environment

```
Get-AdminPowerAppEnvironment -Environment 'EnvironmentName'
```


Which would produce the following detailed information:

```
PS C:\ > Get-AdminPowerAppEnvironment -EnvironmentName fccc53da-0c81-4031-80ea-a93ea68e043a

EnvironmentName      : fccc53da-0c81-4031-80ea-a93ea68e043a
DisplayName           : CDS 2.0 Production (orgd0ed966d)
IsDefault            : False
Location             : unitedstates
CreatedTime          : 2018-07-09T04:52:34.4418252Z
CreatedBy            : @{id=2e804f76-5936-4cf1-bc7c-6faad3d34792; displayName= ; type=User; tenantId=efec
                    userPrincipalName= }
LastModifiedTime     : 2018-07-09T04:52:34.4418252Z
LastModifiedBy      : 
CreationType         : User
EnvironmentType      : Sandbox
CommonDataServiceDatabaseProvisioningState : Succeeded
CommonDataServiceDatabaseType : Common Data Service for Apps
Internal             : @{id=/providers/Microsoft.BusinessAppPlatform/scopes/admin/environments/fccc53da-0c81-4031-80ea-a93ea68e043a; location=unitedstates; properties=}
InternalId           : 
```

Figure 51 Screenshot of PowerShell Get-AdminPowerAppEnvironment output

Another useful command is getting a list of connections in an environment. The following lists all the connections in the tenant's default environment

```
Get-AdminPowerAppEnvironment -Default | Get-AdminPowerAppConnection
```

And finally, a little more complex example. This one pipes the output from one cmdlet to others and presents a nice list of number apps in each environment in the tenant.

```
Get-AdminPowerApp | select -ExpandProperty EnvironmentName | Group | %{ New-Object -TypeName PSObject -Property @{ DisplayName = (Get-AdminPowerAppEnvironment -EnvironmentName $_.Name | select -ExpandProperty displayName); Count = $_.Count } }
```

Which would produce the following detailed information:

```
PS C:\Users\jamesol\Source\Repos\Flow-PowerApps-PowerShell> Get-AdminPowerApp | select -ExpandProperty EnvironmentName | Group | %{ New-Object -TypeName PSObject -Property @{ DisplayName = (Get-AdminPowerAppEnvironment -EnvironmentName $_.Name | select -ExpandProperty displayName); Count = $_.Count } }

DisplayName      Count
-----
Ignite Demo Production    2
Jane Doe's Environment    2
Demo-Build             8
viral trial's Envir...    1
Fabrikam Inc. (defa...    22
Ignite Demo UAT         4
```

Figure 52 Screenshot of PowerShell Get-AdminPowerApp output

CALL TO ACTIONS:

As an administrator you should

- familiarize yourself with the concept of using PowerShell to manage Power Apps and Power Automate in your organization (aka.ms/powerappspowershell)
- download and use our example package for pulling ongoing reports of activity in your environments that can be found via aka.ms/downloadassetsscript

AUTOMATION OF TASKS WITH POWER AUTOMATE

One of the unique things about Power Automate is you can use it to manage itself along with other parts of the Microsoft Power Platform. The following connectors can be helpful to automate administrator tasks with Power Automate.

Connector	Possible uses
Power Automate management connector https://docs.microsoft.com/connectors/flowmanagement/	Can be used to automate working with workflows including getting lists of new workflows or connectors in your environments.
Power Automate for Admins (https://docs.microsoft.com/connectors/microsoftflowforadmins/)	Allows you to perform typical admin actions, such as disabling a flow or deleting a flow
Power Apps for Admins connector (https://docs.microsoft.com/connectors/powerappsforadmins/)	To set permissions on Power Apps or set permissions to a certain connector being used by this app
Power Apps for App Makers connector (https://docs.microsoft.com/connectors/powerappsforappmakers/)	Can be used by the makers themselves, though some actions being an overlay to administrative tasks, such as settings permissions to a Power Apps app – therefore administrators might be using this connector as well
Power Platform for Admins connector (https://docs.microsoft.com/connectors/powerplatformforadmins/)	To perform tasks against platform components, such as creating an environment or provisioning a CDS database or creating a DLP policy for a specific environment
Microsoft 365 (formerly Office 365) Users connector https://docs.microsoft.com/connectors/office365users/	Useful for automating actions around users. For example, you could use the connector to get the manager of a user that owns an environment to be able to send them an email for approval.

Approval connector https://docs.microsoft.com/connectors/approvals/	Often administrators need to get approvals and workflow offers a rich approval set of tasks you can automate this process.
Microsoft Forms https://docs.microsoft.com/connectors/microsoftforms/	Forms is an easy way to collect information to start an admin task. This can be combined with the approval connector to get manager approval.
Azure AD connector https://docs.microsoft.com/connectors/azuread/	Useful to perform tasks such as adding a user to a group or even creating the group.

FOLLOWING, WE WILL PROVIDE YOU SOME TYPICAL EXAMPLES THAT ARE MOST FREQUENTLY ASKED DURING OUR ADMIN-IN-A-DAY WORKSHOPS.GET LIST OF NEW POWER APPS, POWER AUTOMATE FLOWS AND CONNECTORS

One common question is about how to get insights around your citizen app development environment. Who is creating Power Apps apps or Power Automate flows? Which connectors are being used most frequently?

You can solve this task, by simply implementing one of our Power Automate flow templates (<https://aka.ms/listpoweractivity>) we created for you that will provide you a list of new Power Apps, flows and connectors that have been introduced into your tenant within a configurable window. Flow does require Power Apps/Power Automate administrator permissions in order to use the admin connectors.

LIST NEW CONNECTORS CREATED

List new connectors created is a simple template you can get started with right away. It simply triggers daily on schedule and uses the Flow Management connector to get a list of the connection in the environment and sends you an email. You can add it to your flows quickly using the template <https://us.flow.microsoft.com/galleries/public/templates/5a6ef26db3b749ed88b7afb377d11ecf/list-new-microsoft-flow-connectors/>.

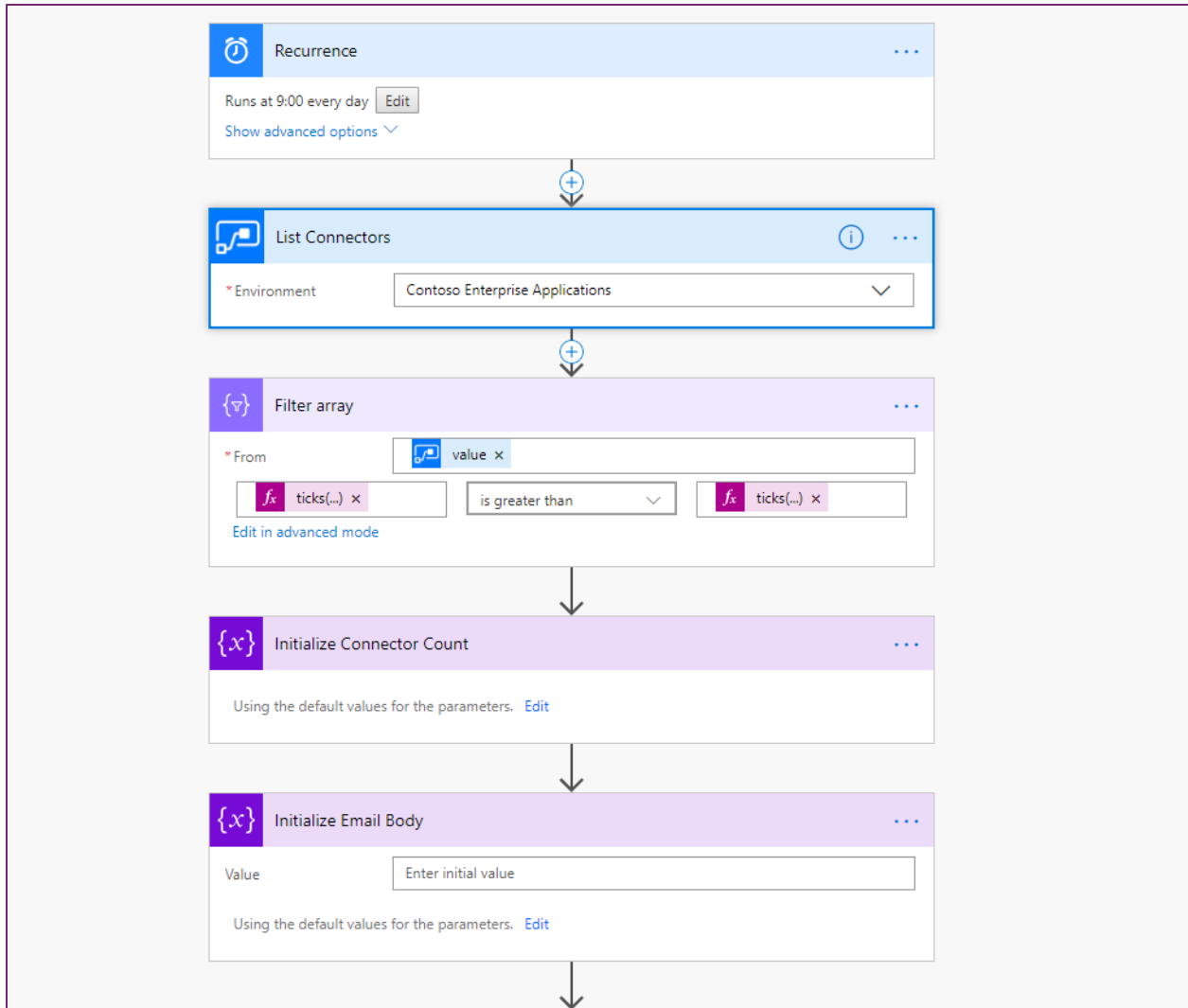


Figure 53: Power Automate flow template for listing connectors

If you want to try building it yourself, there is a good walkthrough of creating the flow from scratch here <https://flow.microsoft.com/blog/new-flow-connector-notifications/>.

WELCOME NEW MAKERS AND IDENTIFY CHAMPIONS

Use this starter template to welcome new makers and point them to resources to be successful. This template <https://aka.ms/powerwelcomeemail> does the following:

- Detect when new workflow has been created.
- Check to see if they are part of makers Azure AD Group.
- If new, send welcome email with company and public resources.
- Invite them to internal Yammer group.

IDENTIFY CHAMPIONS

Use this starter template to get an email sent to you identifying new flows, apps and connector usage in your tenant. The flow template <https://aka.ms/newmakerdigest> enables the following:

- New Power Automate, Power Apps and Connectors Digest
- Sent Daily
- Oversight
- Empower new users

ESTABLISH AND AUTOMATE YOUR AUDIT PROCESS

Use the following templates as examples on how to use the management connectors to permit or restrict behavior based on organization policies.

- **Canvas app, flow creation** – <https://aka.ms/restrictappcreators>
- **Specific connector usage** – <https://aka.ms/restrictflowconnector> ,
<https://aka.ms/restrictappconnector>
- **Newly added connectors** – <https://aka.ms/newconnectornotification>

HOW TO ESTABLISH A PROCESS OF ENSURING ENVIRONMENT CREATION AND APP CREATION (BUSINESS JUSTIFICATION)

Using our Admin-in-a-Day hands on labs (<https://github.com/microsoft/powerapps-tools/tree/master/Administration/AdminInADay>), you will also find an example of how to use Microsoft Forms to establish an approval process around Power Apps apps being created and environments needed for this. To give you an example of how the Power Automate flow could look like:

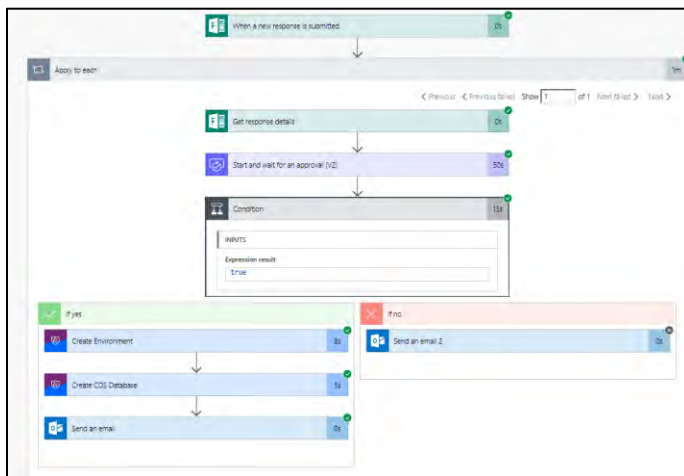


Figure 54: Power Automate flow using Microsoft Forms and Approvals

COE STARTER KIT RESOURCES

The CoE starter kit includes its own audit process that implements the following:

- Apps are identified by a workflow based on criteria such as the app is shared with > 20 Users or at least 1 group and the business justification detail have not been provided.
- Developer Compliance Center where the maker can provide a justification.
- Admin business process workflow for approval.

If you install the starter kit you can use this as it comes with the kit, or tailor it to your own specific requirements.

CALL TO ACTIONS:

As an administrator you should

- familiarize yourself with above examples by installing and testing them in demo environments, before you provision them in your production environment
- make use of our CoE Starter Kit resources (aka.ms/CoEStarterKitDownload) and establish a culture around citizen app development in your organization
- ask Microsoft for partner support around establishing a Center of Excellence in your organization
- ask Microsoft on upcoming events where an Admin-in-a-Day workshop will be provided

RESOURCES TO MANAGE GDPR COMPLIANCE

The European Union General Data Protection Regulation (GDPR) is one of the newest privacy regulations enacted that gives rights to people to manage their personal data. In this section we will look at some of the tools and resources available for the Microsoft Power Platform to assist administrators in their efforts to comply with GDPR. Some of these resources and tools may also helpful to assist you in other data privacy related tasks not directly related to GDPR. A complete discussion of GDPR is beyond the scope of this paper, however in this section we will focus on the tools and resources to support your efforts. Additionally, Microsoft has a section on the trust center dedicated to GDPR resources and information that can be helpful. You can find that here

<https://www.microsoft.com/TrustCenter/Privacy/gdpr/default.aspx> .

First, let's review at some of GDPR's terminology that matters in this context:

Term	Relevance
Data Subject	GDPR identifies people as data subjects. It is their personal data that might have been collected by your organization either in the employment of the

	person or some interaction collecting their personal data.
Data Controller	Organizations that collect and process data for their own purposes.
Data Processor	Organizations that process data on behalf of others.
Personal Data	Any information relating to an identified or identifiable natural person.

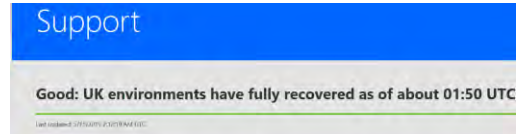
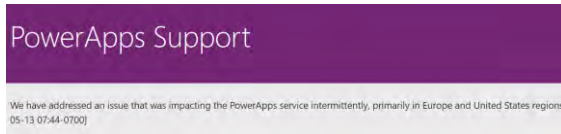
As an administrator one of the key activities in support of GDPR will be related to Data Subject Rights (DSR) requests. These are formal requests from a Data Subject to a Data Controller (likely your organization) to act on their personal data in your systems. GDPR gives rights to Data Subjects to obtain copies, request corrections, restrict processing of the data, delete the data and to receive copies in an electronic format so it could be moved to another Data Controller.

The following links point to detailed information to help you respond to DSR requests depending on the features your organization is using.

Platform Feature Area	Link to detailed response steps
Power Apps	https://docs.microsoft.com/Power Apps/administrator/Power Apps-gdpr-export-dsr
CDS	https://docs.microsoft.com/Power Apps/administrator/common-data-service-gdpr-dsr-guide
Power Automate	https://docs.microsoft.com/flow/gdpr-dsr-summary
Microsoft Accounts (MSAs)	https://docs.microsoft.com/flow/gdpr-dsr-summary-msa
Dynamics 365	https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-dynamics365

SERVICE HEALTH AND PLATFORM UPDATES

Finally, we want to close this section by introducing you to the service health pages for both Power Apps and Power Automate.



<https://powerapps.microsoft.com/support/>

<https://flow.microsoft.com/support/>

In the header section you would find our service health information. In the lower section you would find guidance on learning, common issues, our documentation, link to our communities as well as raising a support ticket, if you inspect something you want to have a follow up with a service engineer on.

Furthermore, we want to outline the importance of the service health dashboard and message center which is part of Microsoft 365 admin center (<https://admin.microsoft.com/>).

Microsoft regularly communicates work done to maintain and update Power Platform and Dynamics 365 services to ensure security, performance, and availability, and to provide new features and functionality.

Microsoft also communicates details of service incidents including the potential user experience, the start and end time of the incident, and any workaround that may be available.

The Service health dashboard is used to communicate service incident information throughout the duration of the event. This would include the following issue statuses:

- **Investigating** – Microsoft is investigating an event to determine the customer impact.
- **Service Degradation/Service Interruption**- Microsoft has confirmed services are being impacted and is taking immediate action to understand the failure and steps required to restore service.
- **Restoring Service** - Cause of the incident has been identified and mitigation steps are in progress.
- **Extended Recovery** - Services are restored but may be slower than usual.
- **False Positive** - Microsoft has confirmed that there was no customer impact

The Message center is used for Proactive communications to inform our customers of any of the following scenarios:

- Deployments (Releases)
- Maintenance
- Customer Action Needed
- Monthly Updates
- Service Changes
- Deprecation of Functionality

DEPLOYMENT, ALM & AZURE DEVOPS

Now that you have read through the [platform architecture section](#) and the data protection concepts and have a good grasp of all the individual components, let's turn to another topic that might be part of a dedicated team in your organization which takes care of deployment and application lifecycle management (ALM) scenarios in general. Nevertheless, we wanted to outline some typical scenarios based on the assumption that you've taken care of the previous suggested creation of some default data loss prevention policies (see suggestions in the compliance and [data protection section](#) of this document). These scenarios represent possible deployment configurations but are not the only ways you could deploy the given scenario. Use them to inspire how you want to handle things in your organization and to understand the general concepts. For more information on these topics please consult the [ALM section of the Power Platform docs](#).

SOLUTIONS

The Common Data Service Solutions Framework provides solutions as containers to track and manage customizations in an environment with CDS. This includes flows, canvas and model-driven apps, entity metadata, forms, views, and other resources required to run the app including developer compiled code assets. A solution starts in the CDS development environment where the app or a flow is created, and the container is used to track any change made to support the app or the flow. Solutions are created and authored by a publisher. Furthermore, solutions allow DevOps engineers taking care of code evaluation, code source control, and other tasks which typically would be referred to as Application Lifecycle Management (ALM) process.

Solutions can be exported from that CDS development environment for transit to other CDS environments. This is commonly used to promote an application from a development environment to test and then finally to a production CDS instance. Exporting a solution and unpackaging them also allows source code evaluation (for instance using solution checker toolkit) and source code check-in to source control systems, such as Azure DevOps.

TYPES OF SOLUTIONS

There are two types of solutions, managed and unmanaged..

- **Unmanaged** solutions are typically to be used in development environments while you are making configuration changes to your application. Solutions are exported as unmanaged and checked into your source control system. Unmanaged solutions should be considered your source.
- **Managed solutions** are used to deploy to any environment outside of development. This includes test, UAT, SIT, and production environments. Managed solutions should be generated by a build server and considered as a build artifact. Though there's a manual UI guided process for creating managed solutions and manually deploying them to different environments. Managed solutions are locked down, meaning you can't directly modify the components. Managed solutions could be manually created by exporting an unmanaged solution and requesting it be exported as managed. That solution when imported into another target CDS

environment is then installed in a managed state. Components in the managed solution can't be directly modified, but they can be added into another unmanaged solution that tracks changes as a separate layer. Multiple managed solutions that are installed in the same CDS instance create layers that combine for what the users see as the effective set of customizations.

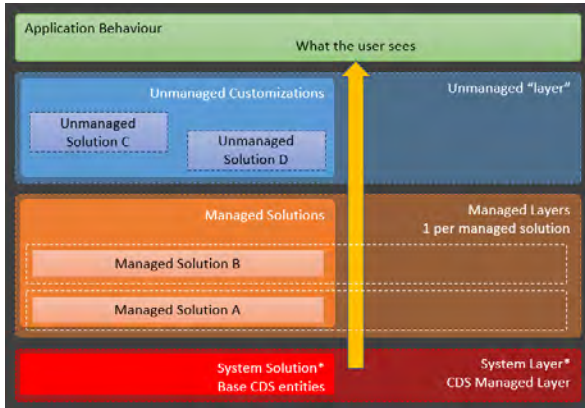


Figure 55: Architecture of solutions

CREATING SOLUTIONS

Each environment has a default solution created automatically as an empty solution when the CDS database is created in the environment. This solution is called *Common Data Services Default solution*.

Directly in the CDS environment you can create additional unmanaged solutions and manage their components using solution explorer.

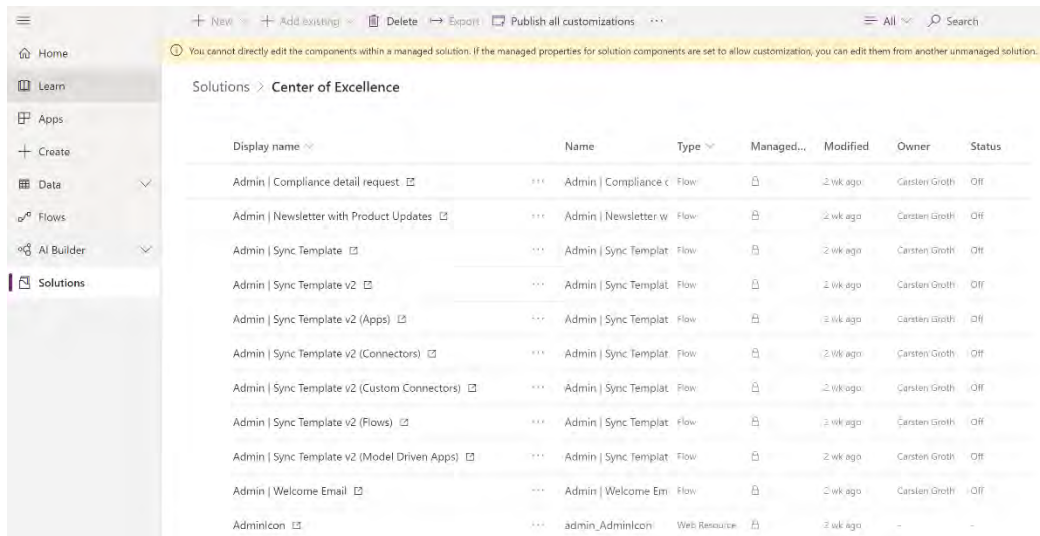


Figure 56: Solution Explorer

INSTALLING SOLUTIONS

Solutions can be installed into an environment with CDS if all their dependencies have been met. A solution becomes dependent when it uses something from another solution. Those dependent solutions must be installed first. Solutions can be installed directly into a target CDS instance from the solution explorer. Solutions can also be deployed using the package deployer tool which can deploy a set of solutions along with data into a CDS instance. Package deployer can be run interactively, or from PowerShell. Package deployer is how Microsoft AppSource marketplace installs apps.

Importing a managed solution is different than importing an unmanaged solution. When you import an unmanaged solution, the changes are merged in with other unmanaged changes in that CDS instance. These merged changes can only be removed by manually removing each item individually. The administrator must also publish the unmanaged changes to have any non-schema (e.g. display labels) changes be visible to other users. Microsoft recommends that unmanaged solutions are only used during development and when installing into test and production environments managed solution should be used.

Solutions also could be installed using Power Apps Build Tools for Azure DevOps. Those can be found in the Marketplace and installed from here. Think of it as the automated way of provisioning your solutions from a Dev/Test environment to your production environment.

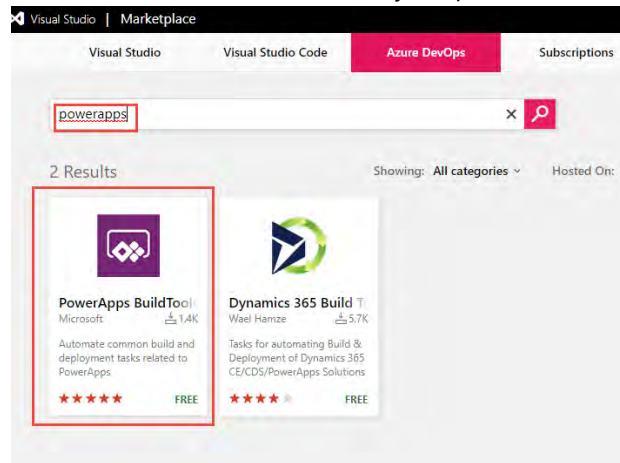


Figure 57: Power Apps Build Tools for Azure DevOps

We will detail more around Azure DevOps in the application lifecycle management section of this document

UNINSTALLING SOLUTIONS

Solutions are uninstalled by deleting them from the CDS environment. The result of the delete action varies greatly between managed and unmanaged solutions. Because unmanaged solutions are merged in with other changes, it is not possible to remove them as a unit. Removing an unmanaged solution simply removes the solution container but all the components that the solution introduced remain in the instance. The remaining components must manually be removed one by one. In fact, some unmanaged changes must be reverted manually such as a label change.

Managed solutions act more like a true uninstall, it removes all the solution components that were installed if nothing new has taken a dependency on them. This includes any data from entities that were only defined and used by that solution being removed. So, take care when removing solutions that you no longer need the data. In many cases you might find that you want to first export the data before the remove/uninstall. Again, this process can be automated with the help of Azure DevOps toolkit.

VERSIONING

For **canvas apps** being saved it creates a new version of the application and it is published for the owner of the application and anyone that has permission to edit the app. Any other user that that application is shared with will still see the "live" version. Once ready, the new version can be published by explicitly clicking on the "Publish this version" link.

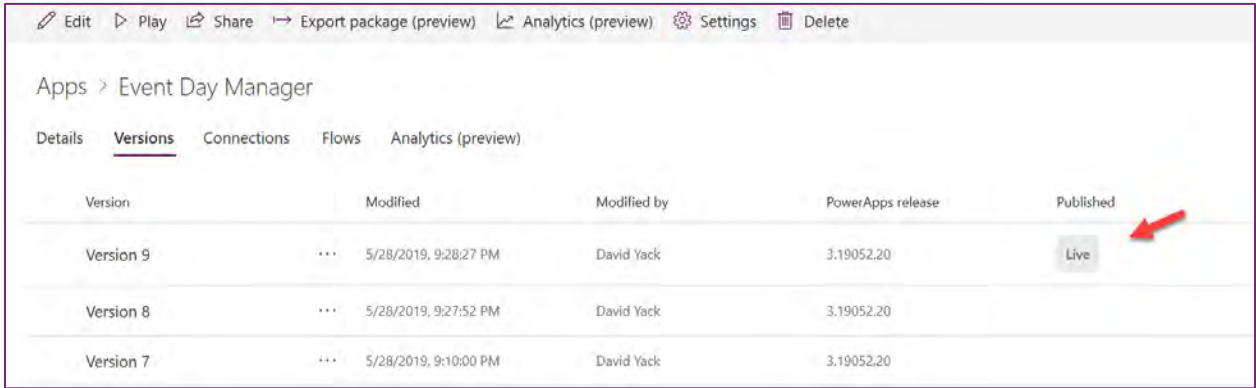


Figure 58: Canvas App Versions

In the event the new version has problems, a prior version can be restored by selecting a prior version and clicking the Restore button next to that version or from the toolbar.

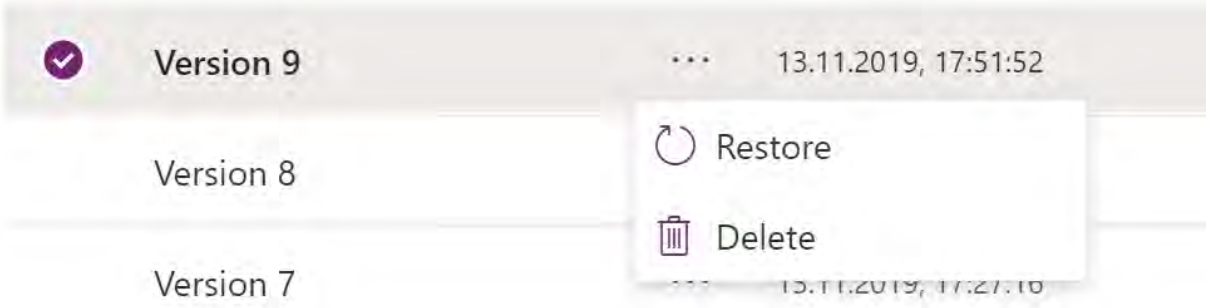


Figure 59: Context menu for restoring a specific version

In the example above, there are nine versions of an app. If the Restore button is clicked on Version 8, Power Apps will create a new Version 10 of the application that is identical to Version 8. In this way history and audit information is preserved and the maker could elect to return to Version 9 and fix issues at a later date. It's important to remember, the published version is the version end users are using. This lightweight application lifecycle management (ALM) is perfect for productivity applications built by your

organization's users without introducing them to the additional overhead of deploying to multiple environments. As apps become more critical to the organization, you should establish more formal application lifecycle management practices. If using this lightweight ALM, we would recommend to also use the version notes which is a convenient way of adding some notes of what has been added or changed in that version and helps in terms of multiple developers acting on the same app. A version note can be maintained, before saving a canvas application.

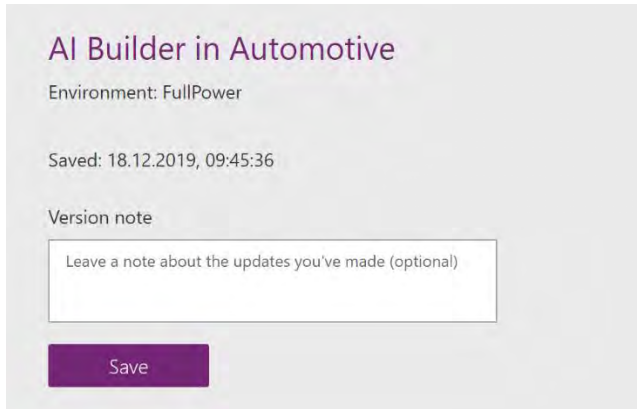


Figure 60: Version note

Version notes can be seen and reviewed by app makers having access to the apps via application details. The versions tab provided here, would offer a column name version note.

Version	Modified	Modified by	Power Apps release	Published	Version note
Version 91	24.10.2019, 13:43:55	Carsten Groth	3.19102.22		
Version 90	23.10.2019, 07:18:04	Carsten Groth	3.19101.25	Live	adjusting barcod...

Figure 61: Version notes under application details

For **model-driven** applications there is also the concept of publish that happens after change of most visual components in the application plus a concept of versioning. Though for model-driven apps the versioning would be done via the solution “container” by either manually cloning the solution to patch and provide a new minor version number.

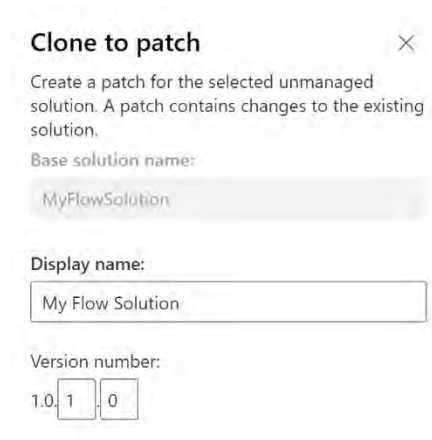


Figure 62: Clone to patch

Or by cloning the solution to a new solution and offering a new major version number as it is shown in the dialog window below.



Figure 63: Clone to solution

For example, if you change the application navigation, users in the same environment will not see the change until publish is completed. Restore is typically accomplished with model-driven applications by exporting a solution version and re-importing it to restore. Therefore, the concept of clone to patch or clone to solution should be understood by every maker working on or with solutions.

Apps and Flows can also be exported into .zip files, see **Appendix to exporting apps and flows**

Application Lifecycle Management (ALM) is important as the applications your organization builds becomes more complex and as more of your company depends on their stability. In other parts of the paper we discussed some of the ALM building blocks that just happen such as versioning of Power Apps canvas apps or versioning regarding solutions. We also covered some of the self-service actions that makers can do such as exporting and importing their CDS solutions. In this section we are going to have a more cohesive discussion about ALM bringing together some of these individual concepts and using them to handle more complex scenarios.

ALM is not a one size fits all concept, it can vary from organization to organization and even within based on the type of solution being built. If you were to look at a typical mission critical solution the following is a good health check of your current Power Platform ALM maturity:

- **Are you deploying managed solutions?** Managed solutions are how Microsoft intends for solutions to be deployed to environments beyond development. All ALM tooling and solution features from Microsoft to support deployment will be targeted towards this goal.
- **Are your development environments single purpose?** As much as capacity allows you should try to have individual development environments for each solution. This ensures you don't get cross solution contamination.
- **Are your development environments disposable?** You should at any point in time be able to easily re-create the development environment. This could be due to someone making corrupting changes or just because you finished development and deleted the old environment and now you're ready to build V2 of the solution. The key to success here is having the unmanaged solution and any dependent managed solutions to import to re-create the environment. Don't forget any reference data that might be needed. Ideally, these assets are stored in source control as we will discuss next.
- **Is source control/Version control your definitive source of truth?** Using a tool like Azure Dev Ops Git repos or another source/version control to track your solution assets allows tracking changes made and by who across releases. While you can check in the whole solution file, this works best in combination with Solution Packager which shards out to a source control friendly and readable format. This also enables you to quickly re-create your dev environment or deploy to production since the solution assets come from the source control repo ensuring a consistent process.
- **Are you using Solution Packager?** Solution Packager allows taking a solution file and breaking it down into individual files for each solution component. This allows what you check into source control to be traced at a very granular level and helps avoid conflicts with multiple people checking in changes. Solution Packager is also how you take individual files from source control

and re-package them for managed solution deployment to other environments like test and production.

- **Can you service (bug fix) production while working on your next version?** A key concept of a healthy ALM practice is not making changes in test or production. By having a good source control and environment strategy you can ensure your dev – test – production release pipeline stays viable even while you are working on the next version.
- **Do you have automated ALM?** While all of the above can be done manually, having an automated repeatable process is ideal. Using the tooling like Power Apps build tools that we will discuss later with Azure Dev Ops much of the ALM process can be automated including the approvals to progress through the release pipeline.

Use the above ALM health check to measure where you are in your goal of having healthy ALM practices for your solutions.

Next, let's look at some of the things you should consider as an administrator to consider helping guide the application through its lifecycles from new to production and then ongoing maintenance and enhancements. For purposes of this section, application refers to the whole set of components from Power Apps canvas or model-driven apps, workflows and any CDS customizations.

New Applications	Existing Applications being upgraded
Who is the application owner, and who is involved in maintaining it?	Are any new connectors being used by the application?
Who are the users of the apps? Are they already licensed?	Is there any new reference data to update?
What environment did you build the app in?	Are there any new canvas, workflows or CDS solutions added in this update?
Are there any Power Apps canvas or model-driven apps as part of the application?	Any changes to how users are assigned security roles?
Are there any workflows?	Any impact on existing CDS data?
What connectors are the apps using?	Any changes in the required licenses?
Does anything require an on-premises gateway?	Potentially any of the considerations from the New Application column, if it was not a consideration at the time.
Does the application use CDS entities?	Are there any ALM automation that is needed?
Is the application dependent on any other existing applications or external services?	

Are there different security roles for different types of users?	
Is there any existing data that must be migrated into the new production system?	
Does the application have reference data that needs to be in the production environment?	
Who will be testing the application? Will it be in a separate environment?	
How will users report problems or enhancements?	
How frequently do you plan to do updates?	
How will ALM be handled?	

The answers to these questions will help you put together an application profile and decide how best to support the team with deploying the application. This is not an exhaustive list, but a starting point for you to develop your own set of questions for applications.

GETTING READY FOR A NEW APPLICATION

Armed with the above information, consider each of the following as you get ready to deploy the new application:

- Licensing – acquire licenses and assign them for users.
- Azure AD Group – consider if having a group that had all the app users would help with sharing the applications components with them. In fact, you might find having a few groups with subsets of the overall application users allows sharing with just the right subset that needs the components.
- Environments – if necessary, create new environments, considering how the application will be tested prior to production deployment.
- Data Loss Prevention policies – do current ones support the app? Are new ones needed? Do you need to adjust for how the application components are using connectors?
- Automation – is there any automation that would help with ongoing app administration?

TOOLS TO HELP MANAGE, PLAN, TRACK, AND DEPLOY

Depending on the complexity of the application, anything from using a SharePoint list to track work to be done and new features, and a OneDrive to store exported assets to a more complete solution like Azure DevOps can help add some structure to your application life cycle process. What is appropriate for your organization depends on the size and maturity of the team that is building the overall application. The less technical team will probably find a solution like OneDrive and SharePoint more approachable. Azure

DevOps has several features that are tailored to support application lifecycle management. Azure DevOps is also free to get started <https://azure.microsoft.com/services/devops/>. The following are some of those features:

- Work item planning and tracking
- Version control – offers a way to store exported assets – using tools like Solution Packager allows this to scale up to larger teams working on CDS Solution package customizations. For more details review <https://docs.microsoft.com/powerapps/developer/common-data-service/compress-extract-solution-file-solutionpackager> .
- Build and release automation – this can be helpful for automating everything from exporting of CDS solutions for backup, to compiling developer-built components. The release automation can take solutions and developer assets and coordinate deploying to test and production environments. These deployments can also leverage approval checkpoints as appropriate. Microsoft has released a preview of Power Apps build tool that include a number of Azure DevOps tasks for automating deployment of CDS Solutions. There are also community tools like Xrm.CI.Framework <https://marketplace.visualstudio.com/items?itemName=WaelHamze.xrm-ci-framework-build-tasks> with which you can deploy CDS solutions .

The following is an example of the Team Status Dashboards that gives the team an all up view of their progress.

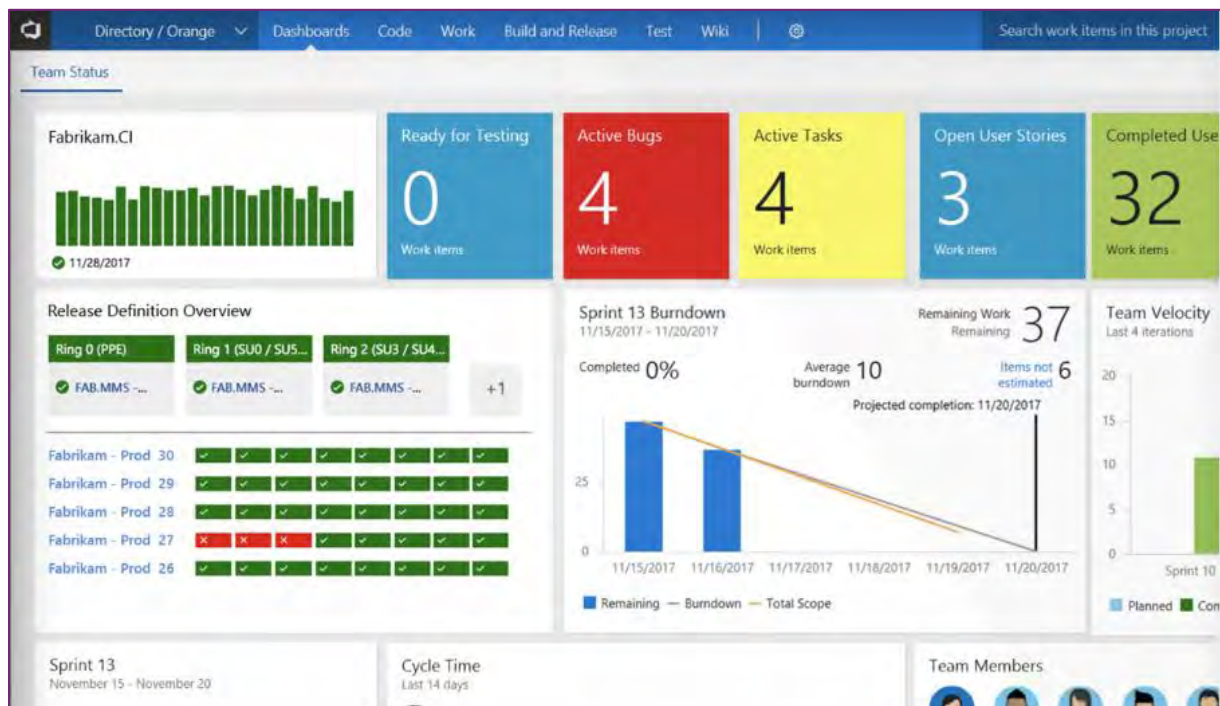


Figure 64: Team status in Azure DevOps

EXPORTING FROM THE SOURCE ENVIRONMENT

We've already covered the concept of exporting from Power Apps, Power Automate and CDS earlier in the document. Let's look at some additional things to consider when exporting as part of an application lifecycle management process.

- Always save a copy of the exported Power Apps canvas app, Power Automate or CDS solution file.
- For CDS Solutions make sure if you are publishing a managed solution, that you also export an unmanaged solution as well. If you are not familiar with the differences, we cover that in the [Platform Architecture section](#).
- For CDS solution export you should always perform a publish on the solution or publish all for all solutions prior to export to ensure all changes are exported as expected. You should also when possible run Solution checker to ensure there are no problems identified.
- For Power Automate flows and canvas apps review the connectors that are used. Any custom connectors will need to be re-created prior to import in the target environment or must be included in the CDS solution.

IMPORTING INTO THE TARGET ENVIRONMENT

We also covered import, but let's look at a few more things to consider:

- Always evaluate what is already in the target environment.
- Create any necessary custom connectors prior to import.
- If you are importing a CDS solution that is dependent on other CDS solutions make sure those are already imported into the CDS instance.
- If you import an unmanaged CDS solution make sure you publish all after import has completed.
- Remember when you import an update to a Power Apps canvas application you must publish the new version before others will see it.
- If you are importing CDS changes that remove any entities and data, consider a proactive on-demand backup prior to the import.

UPDATING EXISTING APPLICATIONS

Shown earlier, the import feature allows the maker to update an existing app in the target environment. Here are some considerations.

- Custom connectors updates must be performed first, as your app may rely on new data definitions.
- Custom connector updates may take a few minutes to be reflected in the portal. During that time, new operations may return a 404 error when invoked.
- If extensive breaking changes are being made, consider creating a new custom connector and leaving the old connector intact. This can also be beneficial in the event the maker needs to roll back, as the previous version of the app will use the old (existing) connector.
- Power Apps uses caching for the web and mobile clients, so changes may not be immediate. For the web client, be sure to clear your cache to see the new changes. On the mobile client, swipe down to refresh app metadata.

ONGOING APPLICATION MAINTENANCE

Once your application has been deployed you can mostly go into maintenance mode responding to user inquiries as needed. Here are a few things to consider while you are between updates.

- Power Apps canvas applications need to be periodically republished for best performance and stability. About every six months you should re-publish your deployed Power Apps canvas applications even if they haven't changed. This ensures the application picks up the latest runtime changes in the environments.
- Keep an eye on your environment storage usage as well as your API quotas and adjust resources and licensing as needed.

RETIRING AND REMOVING AN APPLICATION

As your organization evolves it's likely one or more of the applications deployed will no longer be needed. In fact, you could automate the review of apps that haven't been published in a while and check with the owner to see if it is still needed and if not prepare it for retirement. In this section we will walk through some of the things to consider when retiring an application.

- Confirm that if there are users, they understand the shutdown. Consider shutdown notifications in advance to ensure business continuity and minimize impact.
- Removing access to the application components is often a good first step. Leaving it in this state for a period of time also helps to ensure users know and have a chance to argue their case or

save any data needed.

- Deleting an environment will remove all associated Power Apps, workflows and CDS data. This is not the approach to take if you have multiple applications sharing the environment and you are just retiring a single application.
- When removing connections, you need to first consider the Power Apps canvas apps and workflows that might still be using them. This can be checked by looking at what is associated with the connection prior to deleting.
- Custom connections are sometimes better to be left if they might be reused later as they would require extra effort to re-establish in the future.
- To remove a Power Apps model-driven app depends if the CDS solution containing it was installed as managed or unmanaged. If it was installed as unmanaged you can delete the application module to remove it from users. Removing unmanaged CDS solution components requires manually removing one item at a time from the environment. Removing the CDS solution itself in this situation only removes the container and not the components. This is one of the key benefits of managed solution is the ability to uninstall them as a unit.
- If the solution installed is managed, you would uninstall/remove the CDS solution containing it from the instance. When you remove the CDS solution that contains that application it's important to note that also removes any other components and data as well. If only desiring to remove the application best approach would be to remove the application in the development environment for that CDS solution and then import the update in using the Stage for Upgrade option on import. This will cause only that component to be removed leaving all other components and data intact.

MOVING REFERENCE DATA TO ANOTHER ENVIRONMENT

Often applications have data that is configuration, or reference data. This could be, for example, a list of territories, product lists, or other data that configures and makes the app work. Often components in the application take dependencies on the IDs of this data. The Configuration Migration Tool is designed to move this type of data from one CDS instance to another. The key features of the tool are:

- Select only the entities and fields you for which you want to move data.
- Maintain unique IDs of the records as they are moved.
- Avoid duplicate records by defining a uniqueness condition for each entity based on combination of fields.
- Support updating of existing records.
- Ability to define a schema for what data is moved and use it over and over.

The following outlines the basic process for using the tool.

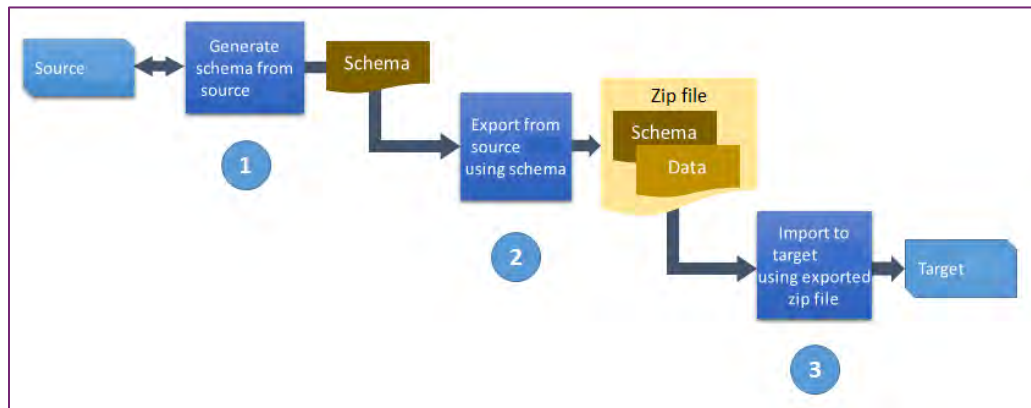


Figure 65: Concept of using Configuration Migration Tool

The output from the tool is a zip file containing the data and the schema file. The same tool can be used to import the data into the target CDS instance. You can also package the data with a solution deployer package that we will discuss shortly allowing it to be deployed alongside one or more CDS solutions. You can read more about how to use the tool here <https://docs.microsoft.com/dynamics365/customer-engagement/admin/manage-configuration-data>.

USING PACKAGE DEPLOYER

So far, we've only talked about importing CDS solutions manually via the user interface. The Dynamics 365 package deployer also works for CDS solutions. The package deployer allows building a package that contains one or more CDS solutions as well as one or more data files to import after the solutions are imported. It is also possible for developers to build custom code that reacts to events from the package deployment process. This code can be used to handle updates to the target environment. Once the package is built, the package can be deployed interactively via the tool, or by command line using PowerShell. You can read more about package deployer here <https://docs.microsoft.com/dynamics365/customer-engagement/developer/create-packages-package-deployer>.

POWER APPS BUILD TOOLS (PREVIEW) FOR AZURE DEVOPS

Power Apps build tools are the building blocks for automating the ALM process using Azure DevOps. They are Azure DevOps tasks that can be included in build and release pipelines to perform the following key actions:

- **General Solution management** - e.g. Set Version, import/export as well as pack and unpack using Solution Packager.
- **Environment management** – This includes creating and deleting environments.

- **Quality Check** – Run Power Apps/Solution Checker inline in a build or release pipeline to check for any problems.
- **Helper tasks** – These include tasks that support the tools like installing the command line tools used by other tasks.

In Azure DevOps these tasks are added by first adding them from the Azure Market place here <https://marketplace.visualstudio.com/items?itemName=microsoft-lsvExpTools.PowerApps-BuildTools> Once added, they are available for use in the build and release pipelines by searching on Power Apps as you see in the following image.

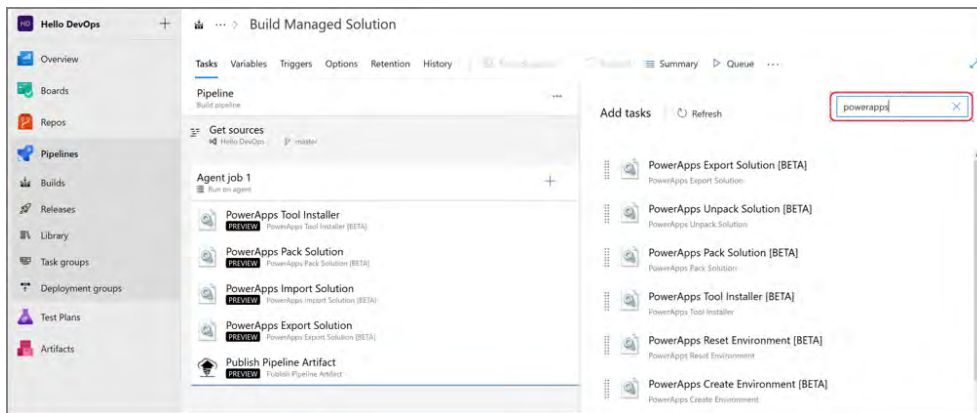


Figure 66: Power Apps Build Tool Tasks

The Power Apps build tools tasks can be used along with any other available Azure DevOps tasks to compose your build and release pipelines. There common pipelines that teams put in place are the Initiate, Build and Release.

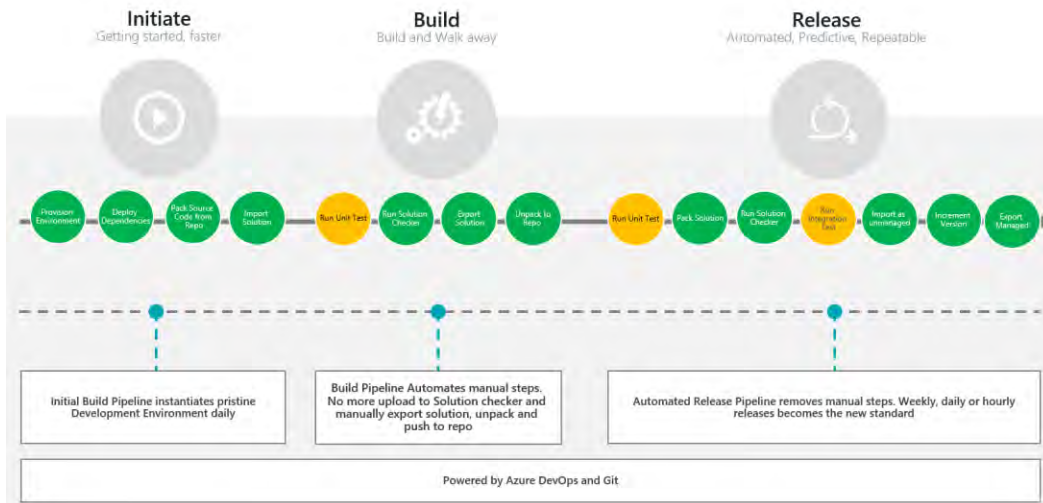


Figure 67: ALM process supported via Azure DevOps

The tools are in preview now and available for trying in your Azure DevOps.

ALM ROADMAP

In the coming months look for Microsoft to release additional tooling for ALM. As a core goal they would like to enable a five-minute to start experience to get setup with an environment with the latest version, have it connected to source control and allow making changes in five minutes or less. This will require improvements to the following areas:

- **Build:** Simplify tooling, consolidate portals and speed up inner loop.
- **Deploy:** Enable an automated repeatable (predictable) deployment methodology.
- **Manage:** Invest in additional environment management capabilities to offer a more flexibility for developers and makers alike to use and dispose preconfigured environments as needed.
- **Monitor:** Application telemetry and feedback loop by design.

To make ALM more approachable the following will be available to help users and teams get started quicker:

- Azure DevOps templates with built-in best practices
- Auto creation of Azure DevOps and Teams projects from PowerApps.com
- Simple UI at PowerApps.com, DevOps for developers / admins
- Single install for tooling
- Pristine and readily available development environments.

The overarching goal is to make good ALM more approachable for projects of all sizes and provide some starting assets to make it easier to follow a healthy ALM process. You can learn more about Application lifecycle management with the Power Platform [here](#).

EDUCATE AND SUPPORT

A broader aspect of managing the Power Platform, and establishing a Power Platform Center of Excellence, is the nurture element to continue growth and onboarding of makers and moving your organization to embrace a digital culture.

While this might not fall under your responsibility as an admin, but below ideas might prove helpful as you adopt the platform further.

Evangelism	Community development	Training and Support
Run internal App / Flow in a day workshops	Create an internal community on Yammer or Teams for your champions	Hold drop-in sessions or regular office hours for your makers to ask questions
Organize hackathons with real business scenarios	Use a Teams or SharePoint site to store resources like your own best practices or brand guides	Provide internal learning resources and tracks for beginner, intermediate and advanced makers
Share success stories	Update your makers about new features in the platform with a monthly newsletter	
Hold Show & Tell sessions to learn what other makers are creating	Offer individual recognition and career paths	

HANDS ON LABS

Hands-on labs for both makers and admins are available to download and include step by step instructions – you can find them here <https://aka.ms/powerplatformlabs>. Below, we are highlighting the most useful labs for admins and makers

For admins

- hands on labs as part of [Admin in a day](#)
- hands on [Power Apps Build tools for Azure DevOps](#) labs

For makers

- [App in a day](#)
- [Flow in a day](#)

App in a day events are frequently offered by partners, you can find a city near you here <https://aka.ms/AIADEvent>

BLOGS

The [Power Apps](#) and [Power Automate](#) teams blog frequently on both new updates as well as ongoing examples of using the features of the products. The Power Automate blog for example has an ongoing series of beginner workflows and intermediate workflows. These are great ways to get ideas even if you don't need that exact solution, it can give you ideas on how to handle similar scenarios. The Power Apps blog has a category for [Admin Features](#).

The blog post on Power Apps Learning Resources - <https://aka.ms/powerappsresources> - is updated frequently, and contains links to learning resources as well as real world customer stories.

COMMUNITY

One of the best resources are the Power Apps and Power Automate community sites:

- The Power Apps community can be found here <https://powerusers.microsoft.com/t5/Power-Apps-Community/ct-p/Power-Apps1>
- The Power Automate community here <https://powerusers.microsoft.com/t5/Microsoft-Flow-Community/ct-p/FlowCommunity> .

These are forums where you can post your question and both the community and Microsoft can respond. Often your issue or question has already been discussed and you can simply look at the prior answers.

SUPPORT TICKET

Support tickets can be created through the Power Platform admin center aka.ms/ppac

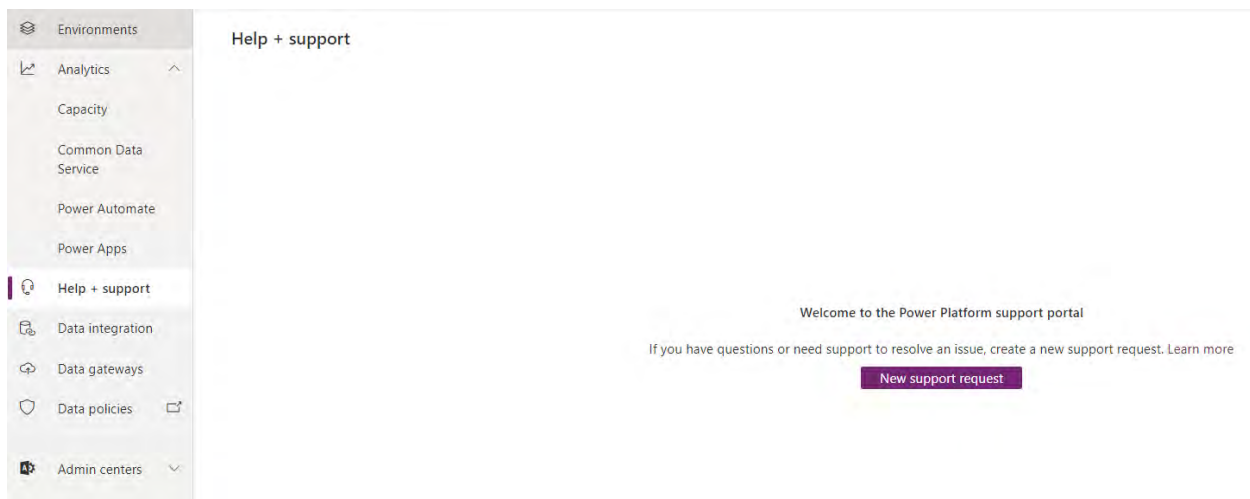
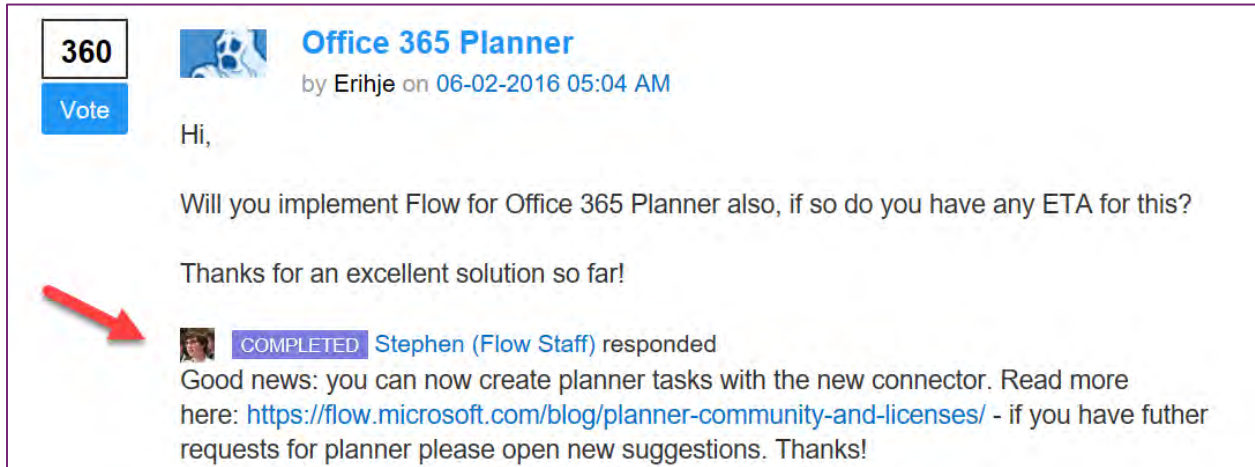


Figure 68 Power Platform Admin Center - Help + Support

SUBMITTING AND VOTING ON IDEAS

The best way to help provide ideas and shape the future of Power Apps and Power Automate is to post your idea or vote on existing ideas to help prioritize them. Power Apps maintains their idea list here <https://powerusers.microsoft.com/t5/Power Apps-Ideas/idb-p/Power AppsIdeas> and Power Automate's list can be found here <https://powerusers.microsoft.com/t5/Flow-Ideas/idb-p/FlowIdeas>. This is also an easy way to see if something has gotten Microsoft's attention or even already completed. Items that can indicate if they are in planning, are in progress by being marked as started, or even already completed.



The screenshot shows a forum post titled "Office 365 Planner" by user Erihje, dated 06-02-2016 05:04 AM. The post has 360 votes, indicated by a "Vote" button. The content of the post is: "Hi, Will you implement Flow for Office 365 Planner also, if so do you have any ETA for this? Thanks for an excellent solution so far!". A red arrow points to a response from Stephen (Flow Staff), which is marked as "COMPLETED". The response text is: "Good news: you can now create planner tasks with the new connector. Read more here: <https://flow.microsoft.com/blog/planner-community-and-licenses/> - if you have futher requests for planner please open new suggestions. Thanks!".

Figure 69 Power Platform Ideas Forum

MICROSOFT LEARN

Microsoft Learn offers short courses that can be consumed by both makers and administrators. The Power Apps courses can be found here <https://aka.ms/powerup>. Administrators will likely find the managing application courses a good fit. For Power Automate you can find the courses here <https://docs.microsoft.com/flow/guided-learning/>. There are also a couple courses on administering workflows would be good for administrators.

FINDING CONSULTING PARTNERS

If you find that you or your teams are looking for some outside assistance you can use the Partner Finder to locate a partner that specializes in Power Apps and Power Automate. You can find the list of partners here <https://PowerApps.microsoft.com/partners/>. The Partner Showcase is also a good place for inspiration as well as to take a look at some of the amazing things partners have built on the platform. You can find the showcase here <https://PowerApps.microsoft.com/partner-showcase/>.

NEXT STEPS

Congratulations, you've made it to the end! We hope you found some helpful information and keep this paper around for future reference. To remind you of where we started our journey at the beginning of the document, here are the actions you should consider taking in the first year to get started on the right track!

- Identify the central team that will be implementing Power Platform governance and assign them the [Power Platform service admin role](#), which grants full access to PowerApps, Flow, & Power BI
- Establish an environment strategy, restrict the creation of net-new trial and production environments to admins, and automate a process for requesting new environments
- Setup data loss prevention policies
- Leverage out-of-box activity logs & analytics
- Don't start from scratch, learn from the Center of Excellence starter kit
- Establish and automate your audit processes
- Welcome new makers and identify champions
- Establish a Center of Excellence that will help accelerate your adoption of the platform by investing in and nurturing organic growth while maintaining governance and control. Your Center of Excellence will be aligned to and drive your company's digital transformation strategy and goals.

APPENDIX

APPENDIX TO ENVIRONMENT STRATEGY

IMPACT OF MULTIPLE ENVIRONMENTS ON USERS

make.powerapps.com and flow.microsoft.com display one environment only. By default, that environment will be set to the tenant default environment. Users can change their environment in the players and portals using the environment selector.

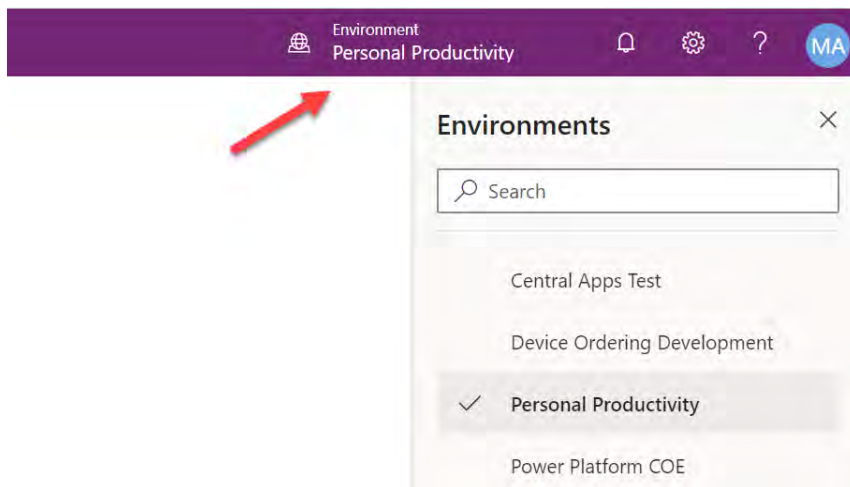


Figure 70 Environment toggle on make.powerapps.com

A maker whose applications and assets are distributed across multiple environments will have to adjust the environment settings to continue working on their assets.

An end user using the Power Apps mobile applications is presented with a consolidated list of applications across the tenant environments they have access to. Each application indicates the environment. This reduces the need to switch, however it introduces the need for the user to choose the correct application. For example, imagine if you had an application *Device Ordering* and it was deployed to environment Test and environment Production. If the user had access to both environments it would show up twice on the list. The user would have to differentiate between the two. Some of this can be minimized by only granting access as needed and then only temporarily to the test environment.

IMPACT OF MULTIPLE ENVIRONMENTS ON CONNECTORS

When an application uses a public connector (available for all tenants), the connection to the connector is configured for use within the context of one environment. Custom connectors are also configured in the context of one environment. If an app is moved to another environment the public connector references will be recreated upon import. Custom connectors must be re-configured manually in that target environment or transported via a solution.

Applications that use the Common Data Service connector have the option of connecting to the current environment or a specific environment. Selecting current works well for apps that need to move between dev, test and production instances because it adjusts automatically when imported into the next environment. Applications that use the Common Data Service (Current Environment) connector are pre-configured to point to the current environment without the option of choosing a specific environment.

IMPACT OF MULTIPLE ENVIRONMENTS ON CDS

When thinking about how to organize your environments you should consider where your data lives. Having multiple environments, each with their own CDS database, might make sense in a few different scenarios. First, users have data that is geographically separated, and they don't share across those boundaries. Second, data from different applications that have conflicting incompatible use of CDS. Third, where users are building personal or team productivity applications that need CDS data but as an organization you aren't ready to mix that with the rest of your enterprise data.

APPENDIX TO RESOURCE SHARING

USER ACCESS TO APPS

Users obtain access to apps by having them shared with them. The technical specifics of how that sharing works is different between canvas apps and model-driven apps.

For **canvas apps** they are shared with users, Azure Active Directory Security Groups or with the whole organization.

Model-driven apps you share by adding a user to a CDS security role that is associated with the application. You can now also leverage Azure AD Groups associated with CDS teams to accomplish this. Users of the Azure AD group are automatically added to the CDS team as members. The team is assigned a security role and the user gets access to the app and data by being a team member. You can read more on CDS group teams here <https://docs.microsoft.com/power-platform/admin/manage-teams#about-group-teams>. We will cover more on CDS security roles in the [security section](#) of this paper.

The below is a screenshot of sharing a canvas app; when you share a canvas app, the user will also need access to the resources the app depends on. Some of the resources are shared implicitly, others require that the user logins with their own credentials to use the connection. From this sharing panel you can also associate a security role that grants access to CDS data used by the app.

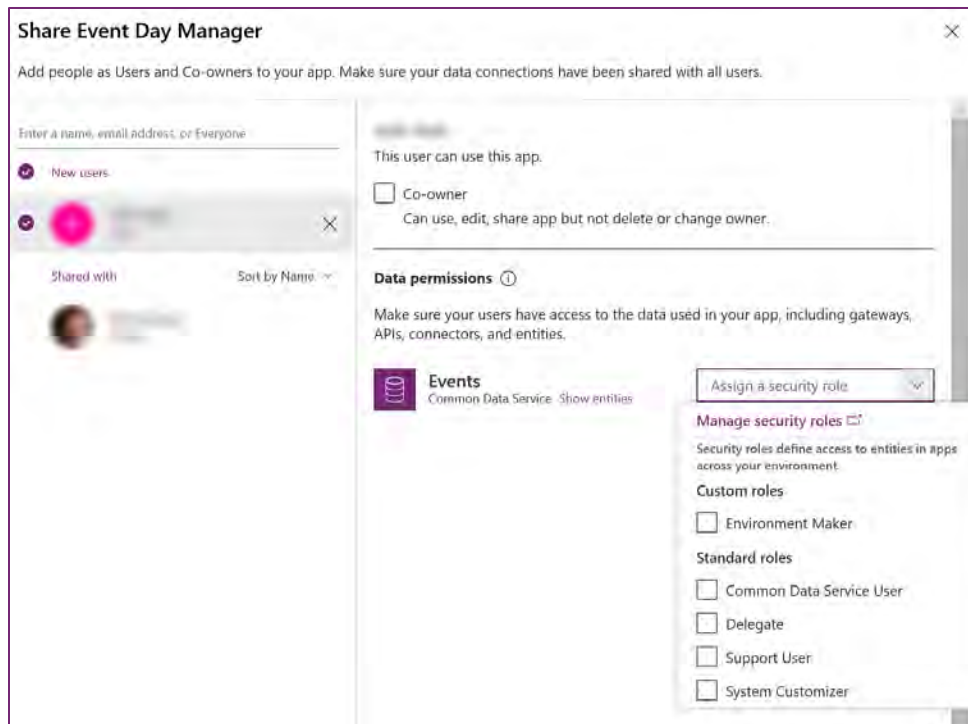


Figure 71 Canvas app sharing with CDS Security Roles

APPLICATION PLAYERS

Both types of applications can be used as web applications from mainstream web browsers. Both types of applications can be discovered from make.powerapps.com. Model-driven apps can also be discovered by going to the environment url e.g. environmentname.crm.dynamics.com.

End users can discover applications through home.dynamics.com application list as well as in the common application navigation list through the Microsoft 365 (formerly Office 365) hamburger menu. There is no dedicated landing zone for Power Apps apps, end-users can access them via home.dynamics.com even if you only have a Power Apps and not a Dynamics 365 license.

Mobile users can run the application in a device installed player app on both phones and tablets. Users will login to the mobile app using their Azure AD credentials. Power Apps mobile apps on Android and iOS are integrated with Microsoft Intune and support Intune policies for mobile application management. Currently, the player application for canvas apps is different from model-driven apps. To run a Power Apps model-driven app from a mobile device install the Dynamics 365 app from the app store.

SHARING OF CANVAS APPS THAT USE CONNECTORS

Some types of connections, such as SQL Server, are shared automatically, but others require users to create their own connections to the data source or sources in the app.

On powerapps.com, you can determine whether a connection will be shared automatically, and you can update sharing permissions. In the left navigation bar, click or tap Manage, click or tap Connections, and then click or tap a connection. If the Share tab appears, the connection will be shared automatically.

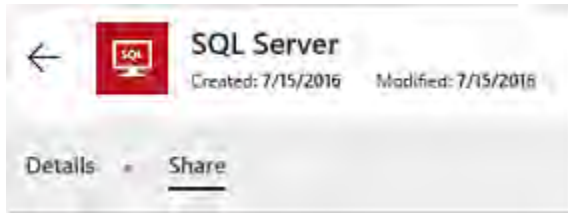


Figure 72 SQL Server connection

If it is then the connection will be shared implicitly. Otherwise, the user will need to create their own connection. Custom connectors are shared, but users must create their own connection to it. This means that the user the connector is shared with needs to have credentials or a key if required by the custom connector. A custom connector connection, however, can be explicitly shared giving the user the right to use, use and share or edit the connection. More information on custom connectors can be found here <https://docs.microsoft.com/connectors/custom-connectors/>

SHARING OF POWER AUTOMATE FLOWS THAT USE CONNECTORS

Power Automate flows can be shared with other users either as co-owners or run-only users. When a user adds another user or group as an owner of a workflow those users will have full access to all the connections used in the workflow. This means if they run the workflow it will take the action in the context of the user signed into the connection. Because they are co-owners of the workflow, they will also be able to modify it using the connections that already exist. They may also change the login on the connection. However, they are not required to do so. Co-owners are limited to use the connection with that workflow. They can't create a new workflow and use the same connection. The following is an example of the warning that is presented when you add a co-owner.

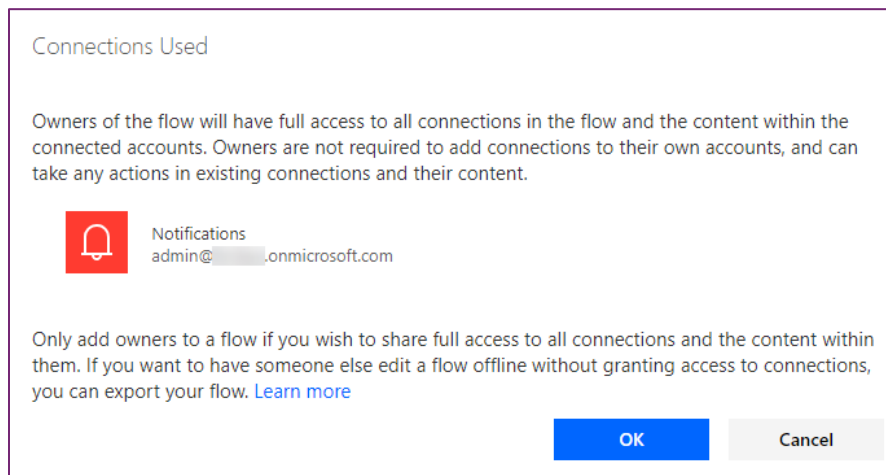


Figure 73 Power Automate flow - co-owner warning

A best practice is to ensure an account with minimum permissions is used when setting up a flow. For some connectors like CDS and SQL Server you can also use a service principal which allows the connector to access the service in the context of the application identity and not a traditional user account.

Run-only sharing is an option when the flow is manually triggered. This option allows greater control because first of all the user does not have ability to edit the flow, just the ability to run it. Second, when you invite the user you can specify to reuse the existing connection or require the user to provide their own. To manage the run-only users drill down on the flow from the list of flows and you will see the following:

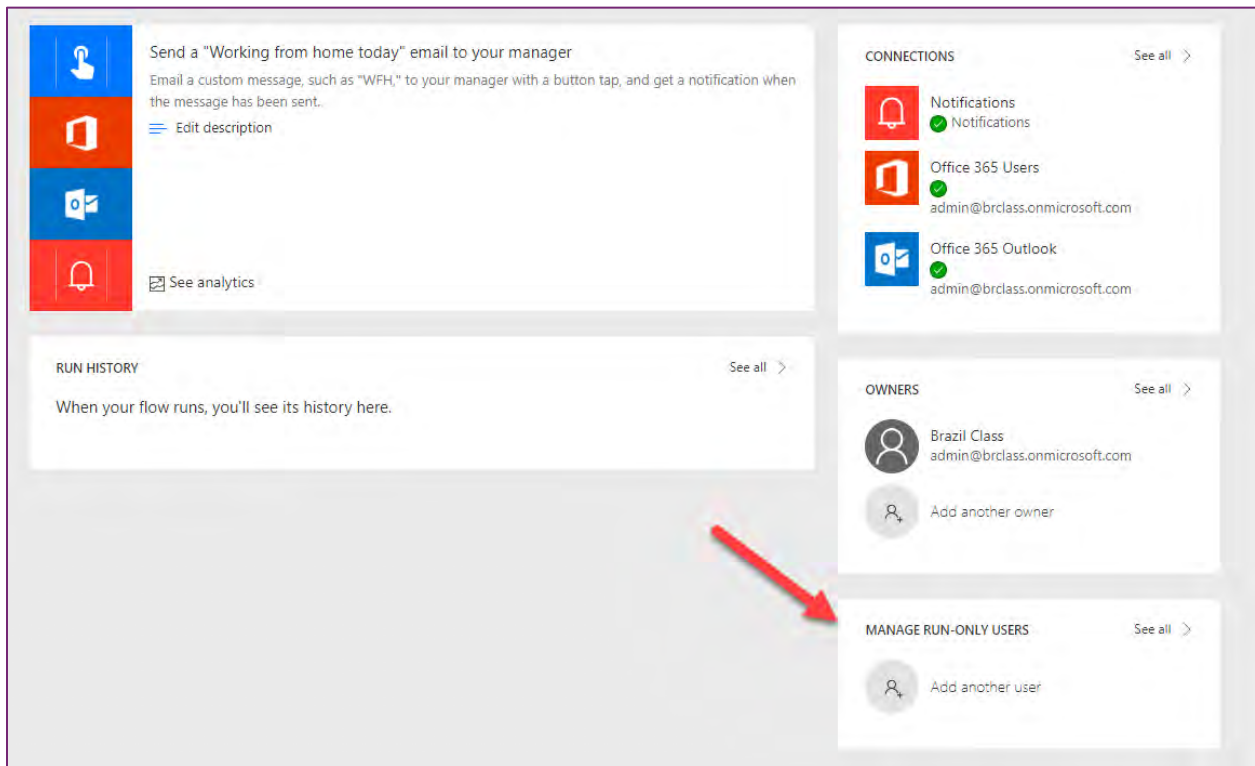


Figure 74 Power Automate flow - Manage run-only users

From here you will see a dialog to specify the user or group as well as a list of the connections and the choice for each on how to grant access. The following shows the connection configuration and how you can choose to force the user to sign-in to their own connection.

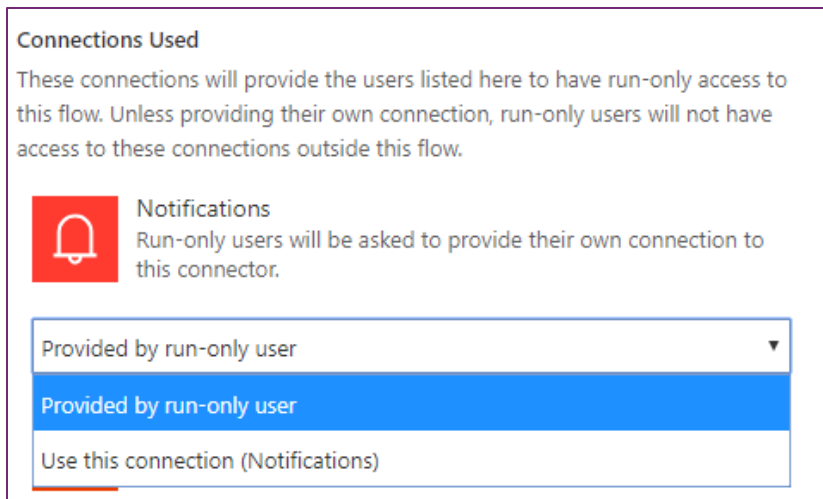


Figure 75 Sharing a flow - specify connections used

One of the more recent additions is the ability to share a flow with a SharePoint list or an Microsoft 365 (formerly Office 365) Group. In this scenario, the flow is available to all members of the group. For SharePoint lists, anyone with edit access to the list would have access to the flow. The flow would then show up with the ability to execute it from the application navigation.

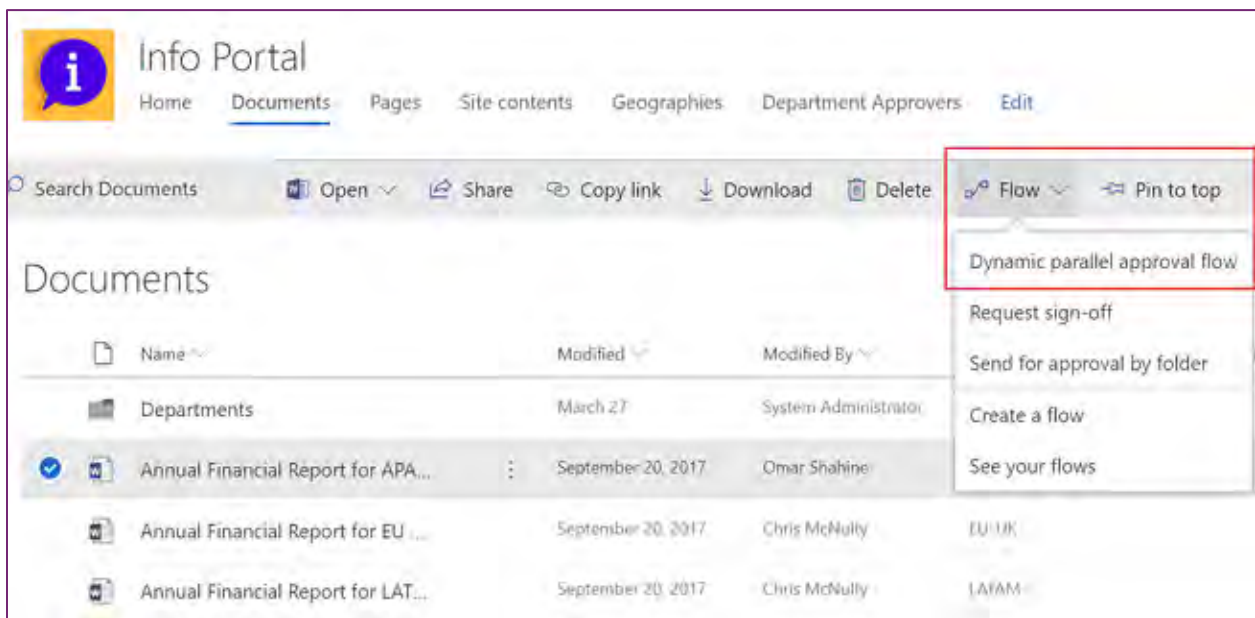


Figure 76 SharePoint Flow

CONNECTOR AUTHENTICATION PATTERNS

Power Apps and Power Automate authenticate with connectors to create a connection instance. It is that instance that contains the specific configuration information necessary for the app or workflow to talk to the connector API that is used in each interaction. Connectors could choose to use no authentication, basic authentication, API key authentication or OAuth 2.0. The most common are OAuth and API Key.

If you aren't familiar with OAuth, it is an authorization framework that allows external applications to obtain controlled access to a target service. Many APIs support it including CDS, Facebook and Twitter, to name a few. The goal of authentication is to allow the user to sign into a familiar login dialog, consent to the application using the service, and then setup to allow tokens to be acquired. It is the tokens that are used on each request to prove who the user is and their right to use the API. In the Power Apps and workflow usage, a consent server is involved that helps manage the tokens and their lifecycle including storing the renewal token in the consent server and handling the refresh cycle. The following is a step by step look at what happens when you authenticate a connection using OAuth.

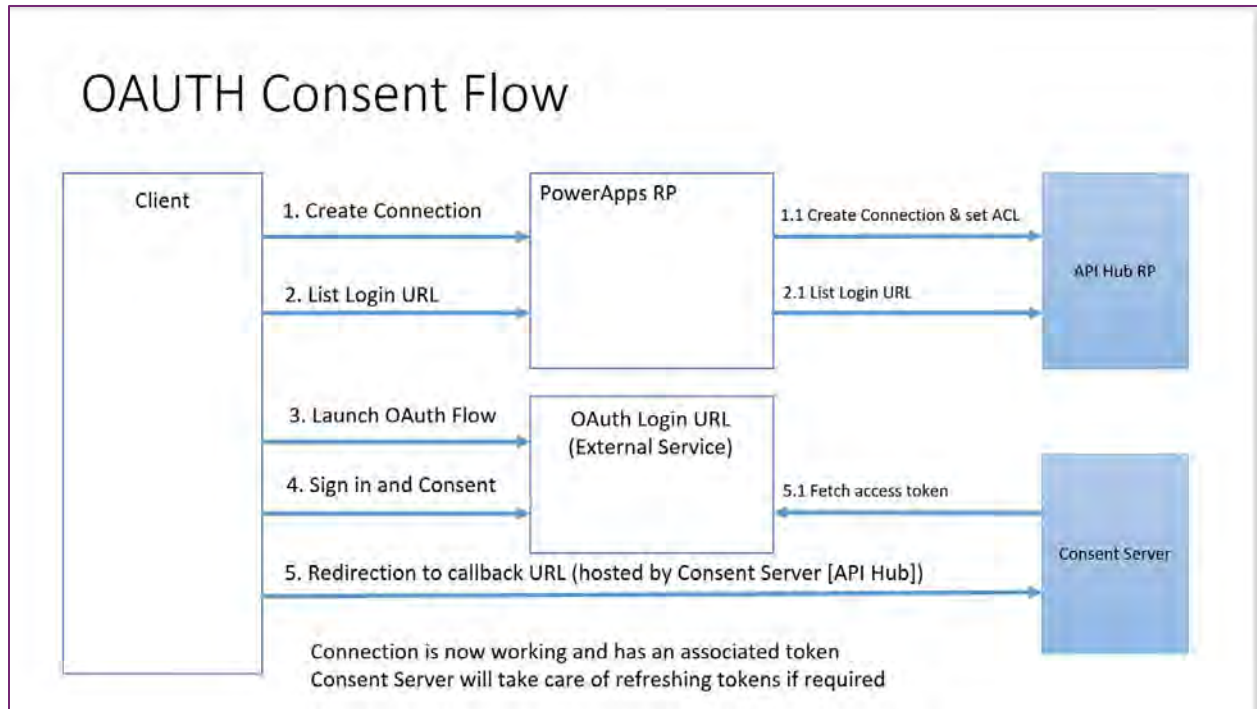


Figure 77 OAuth Consent Flow

The API key is a little less complex as it typically involves the API assigning a key that is passed on each request. That key is provided when the connection is established for the connector and is stored in the environment with the other connection information in a secure way. An example of an API key authentication connector is the Azure Storage Blob. As you can see below it wants the Storage Account Name as well as the Access Key.

Figure 78 shows a configuration dialog for an Azure Blob Storage connector in Microsoft Flow. The dialog is titled "Azure Blob Storage - When a blob is added or modified (properties on...". It contains three required fields, each marked with an asterisk:

- * Connection Name**: A text input field with the placeholder text "Enter name for connection".
- * Azure Storage Account name**: A text input field with the placeholder text "Name of the storage account the connector should use."
- * Azure Storage Account Access Key**: A text input field with the placeholder text "Specify a valid primary/secondary storage account access key."

At the bottom center of the dialog is a "Create" button.

Figure 78 Flow action requesting API key

APPENDIX TO ON-PREMISES DATA GATEWAY

GATEWAY NOVEMBER 2019 UPDATES

On-premises data gateway are supported [in multiple environments](#) and [multiple regions](#).

For a full list of November 2019 updates for on-premises data gateway go to <https://flow.microsoft.com/blog/on-premises-data-gateway-november-2019-update-is-now-available/>

GATEWAY ON-PREMISE INSTALL

The gateway service must run on a local server in your on-premises location. It is not recommend that the server is the same one as the resources it will proxy access to, however it should be on the same local network to reduce latency. It does need to be able to access the target resource with as low of latency as possible. Multiple application and workflow connections can use the same gateway install. You can only install one gateway on a server.

During the install the gateway is setup to use NT Service\PBIegwService for the Windows service logon. You can switch this to a domain user or managed service account if you'd like.

GATEWAY ADMINISTRATION ACCESS

By default, you have this permission on any gateway that you install. As the administrator you can grant other users or Azure AD group permission to co-administrate the gateway. It is recommended you always have multiple administrators specified to handle employee events in your organization.

From the [Power Platform admin center](#) you can now see a list of all the gateways and manage who has access to install gateways.

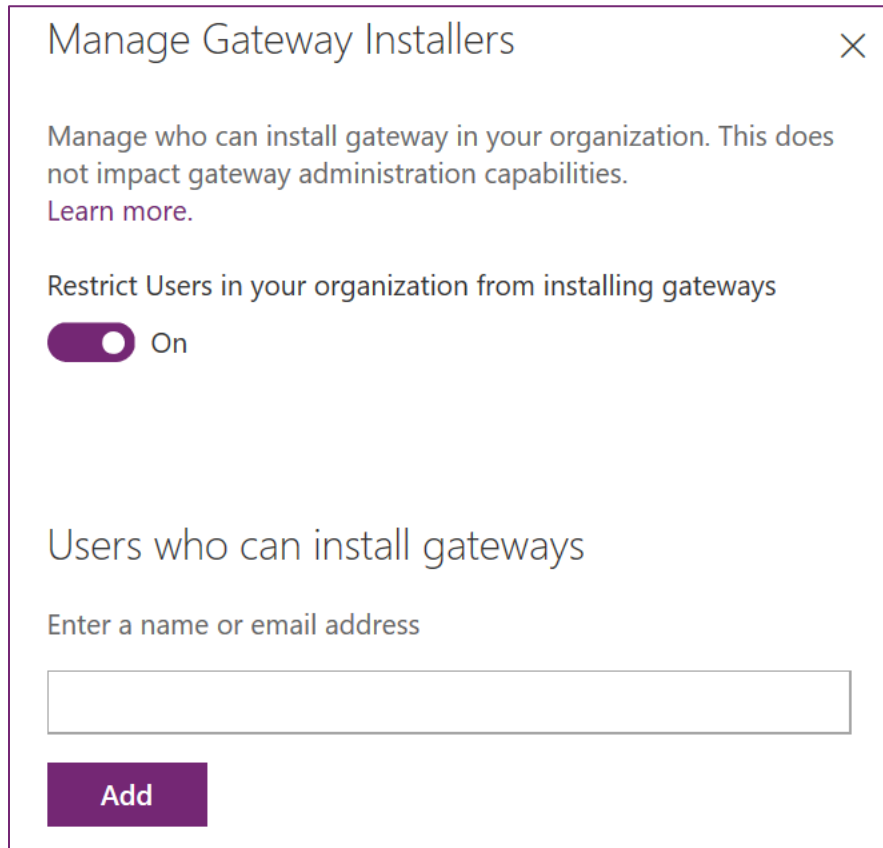


Figure 79 Gateway Installers

USE OF STORED CREDENTIALS

When you setup a data source on the gateway you will need to provide credentials for that data source. All actions to that data source will run using these credentials. The credentials that you enter for a data source are encrypted and stored in the gateway cloud service. The credentials are decrypted at the gateway on premises. The credentials are sent to the machine running the gateway on-premises, where they are decrypted when the data source is accessed.

PORT USAGE

The gateway service creates an outbound connection to Azure Service Bus so there are no inbound ports required to be open. The outbound connection communicates on ports: TCP 443(default), 5671, 5672 9350 through 9354.

It is recommended that you whitelist the IP addresses for the data region in your firewall. You can download the latest list here <https://www.microsoft.com/download/details.aspx?id=41653> . These IP addresses are used for outbound communication with Azure Service Bus.

UPDATES TO THE DATA GATEWAY

Updates are not auto-installed for the on-premises data gateway. It is highly encouraged to remain current with the latest data gateway version as the updates to the gateway is are released on a monthly basis.

GATEWAY DISASTER RECOVERY

A recovery key is assigned (i.e., not auto-generated) by the administrator at the time the on-premises data gateway is installed. The recovery key is required if the gateway is to be relocated to another machine, or if the gateway is to be restored. Therefore, the key should be retained where other system administrators can locate it if necessary.

APPENDIX TO CDS SECURITY ROLES

BUSINESS UNITS

Business units work in conjunction with security roles to determine the effective security that a user has. Business units are a security modeling building block that helps in managing users and the data they can access. Business units define a security boundary. Every CDS database has a single root business unit. You can create child business units to help further segment your users and data. Every user assigned to a CDS instance will belong to a business unit. While business units could be used to model a true organization hierarchy, more often they lean more towards defined security boundaries to help achieve the security model needed.

To better understand let's look at the following example. We have three business units. Woodgrove is the root business unit and will always be at the top, that is unchangeable. We have created two other child business units A and B. Users in these business units have very different access needs. When we associate a user with this CDS instance, we can set the user to be in one of these three business units. Where the user is associated will determine which business unit owns the records that user is the owner of. Having that association allows us to tailor a security role to allow the user to see all records in that business unit.

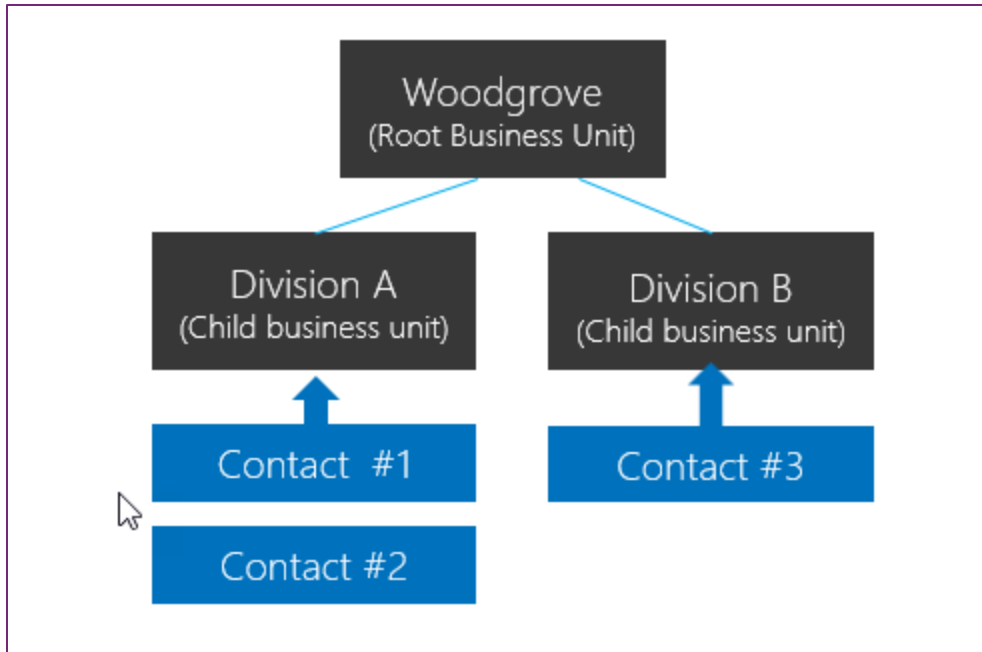


Figure 80 CDS Security Roles - Business Units

ENTITY/RECORD OWNERSHIP

CDS supports two types of record ownership; organization owned, and user or team owned. This is a choice that happens at the time the entity is created and can't be changed. For security purposes, records that are organization owned, the only access level choices are either the user can perform the operation, or the user cannot. For user and team owned records, the access level choice for most privileges are tiered Organization, Business Unit, Business Unit and Child Business Unit or only the user's own records. That means for read privilege on contact, I could set user owned, and the user would only see their own records.

To give another example, let's say User A is associated with Division A, and we give them business unit level read access on contact. They would be able to see Contact #1 and #2 but not Contact #3.

When you configure or edit security role privileges you are setting the access level for each option. The following is an example of the Security Role privilege editor.

Entity	Create	Read	Write	Delete	Append	Append To	Assign	Share
Account	●	●	●	●	●	●	●	●
ACIViewMapper	●	●	●	●				
Action Card	●	●	●	●	●	●	●	
Action Card User Settings	👤	👤	👤	👤				👤
Activity	●	●	●	●	●	●	●	●
Announcement	●	●	●	●		●		
Application File	●	●	●	●				
Azure Service Connection	●	●	●	●	●	●		
Connection	●	●	●	●	●	●	●	●
Connection Role	●	●	●	●	●	●		
Contact	●	●	●	●	●	●	●	●
Customer Relationship	●	●	●	●	●	●	●	●

Figure 81 Security Role privilege editor

In the above you can see the standard privilege types for each entity Create, Read, Write, Delete, Append, Append To, Assign and Share. You can edit each of these individually. The visual display of each will match the key below as to what level of access you have granted.

Key									
○	None Selected	👤	User	👤	Business Unit	●	Parent: Child Business Units	●	Organization

Figure 82 Standard privilege types

TEAMS

Teams are another important security building block. Teams are owned by a business unit. Every business unit has one default team that is automatically created when the business unit is created. The default team members are managed by CDS and always contain all users associated with that business unit. You can't manually add or remove members from the default team, they are dynamically adjusted by the system as new users are associated/disassociated with business units. There are three types of teams; owning teams, access teams and Azure AD group teams. Owning teams can own records, which gives any team member direct access to that record. Users can be members of multiple teams. This will allow it to be a powerful way of granting permissions to users in a broad way without micromanaging access at the individual user level. Access teams are discussed below as part of record sharing.

Azure AD group teams are a type of team similar to owning teams but connected to an Azure AD Security or Office group. With this type of team, the members of the CDS team are synchronized with Azure AD automatically. You can manually create these teams and connect them to Azure AD by providing the existing Azure AD group ID. They are also created automatically if you share a Power Apps canvas app with an Azure AD group and then choose to assign a security role. As a result of that share the system will create the team, hook it up to the Azure AD group and then associate the chosen security roles with that CDS team. While it is possible to assign the System Administrator security role to a team because the role

is not configured for direction assignment, this configuration will not produce the same result as assigning System Administrator directly to a user.

RECORD SHARING

Individual records can be shared on a one by one basis with another user. This is a powerful way of handling exceptions that don't fall into the record ownership or member of a business unit access model. It should be an exception though because it is a less performant way of controlling access. Sharing is more difficult to troubleshoot because it is not a consistently implemented access control. Sharing can be done at both the user and team level. Sharing with a team is a more efficient way of sharing. A more advanced concept of sharing is with access teams which provides auto-creation of a team and sharing of record access with the team based on an access team template (template of permissions) which is applied. Access teams can also be used without the templates, with just manual add/remove of its members. Access teams are more performant because they don't allow owning records by the team or having security roles assigned to the team. Users get access because the record is shared with the team and the user is a member.

RECORD LEVEL SECURITY IN CDS

You might be wondering – what determines access to a record? That sounds like a simple question but for any given user it is the combination of all their security roles, the business unit they are associated with, the teams they are members of and the records that are shared with them. The key thing to remember is all access is cumulative across all those concepts in the scope of a CDS database instance. These entitlements are only granted within a single database and are individually tracked in each CDS database. This all of course requires they have an appropriate license to access CDS.

FIELD LEVEL SECURITY IN CDS

Sometimes record level control of access is not adequate for some business scenarios. CDS has a field level security feature to allow more granular control of security at the field level. Field level security can be enabled on all custom fields and most system fields. Most system fields that include personal identifiable information (PII) are capable of being individually secured. Each field's metadata defines if that is an available option for the system field.

Field level security is enabled on a field by field basis. Access is then managed by creating a field security profile. The profile contains all fields that have field level security enabled and the access granted by that specific profile. Each field can be controlled within the profile for create, update and read access. Field security profiles are then associated with a user or teams to grant those privileges to the users to the records to which they already have access. It's important to note that field level security has nothing to do with record level security. A user must already have access to the record for the field security profile to grant them any access to the fields. Field level security should be used as needed and not excessively as it can add performance overhead that is detrimental if overused.

MANAGING SECURITY ACROSS MULTIPLE ENVIRONMENTS

Security roles and field security profiles can be packaged and moved from one environment to the next using CDS solutions. Business units and teams must be created and managed in each CDS environment along with the assignment of users to the necessary security components.

CONFIGURING USER'S ENVIRONMENT SECURITY

Once roles, teams and business units are created in an environment it is time to assign the users their security configurations. First, when you create a user you will associate the user with a business unit. By default, this is the root business unit in the organization. They are also added to the default team of that business unit.

In addition, you would assign any security roles that user needs. You would also add them as members of any teams. Remember teams can also have security roles, so the effective rights of the user is the combination of directly assigned security roles combined with those of any teams they are members of. As we discussed earlier you could use the Azure AD team feature to automatically create the CDS team, associate security roles and synchronize the team's members based on the Azure AD group members. Security is always additive offering the least restrictive permission of any of their entitlements. The following is a good walkthrough of configuring environment security <https://docs.microsoft.com/PowerApps/administrator/database-security>.

If you have used field level security, you would also need to associate the user or a team of the user to one of the field security profiles you created.

APPENDIX TO SHARING APPS IN TEAMS

Sharing and managing apps and flows in Teams

Sharing apps: Currently access permissions for Power Apps embedded in Teams are managed through the Power Apps UI using the sharing process outlined earlier. For example, once an app is pinned to a channel, the app owner must ensure that the app has been shared with the individual members or the entire channel from the Power Apps UI.

Sharing flows: Flows can be shared with individuals, channels, or teams from within Teams. To share these flows, open the Power Automate app (formerly the Flow app) in Teams, select the desired app to share, click the "share" button, then enter the individuals, channel, or team to share with. If the flow had previously been listed under "my flows", it will move to the "team flows" list after sharing.

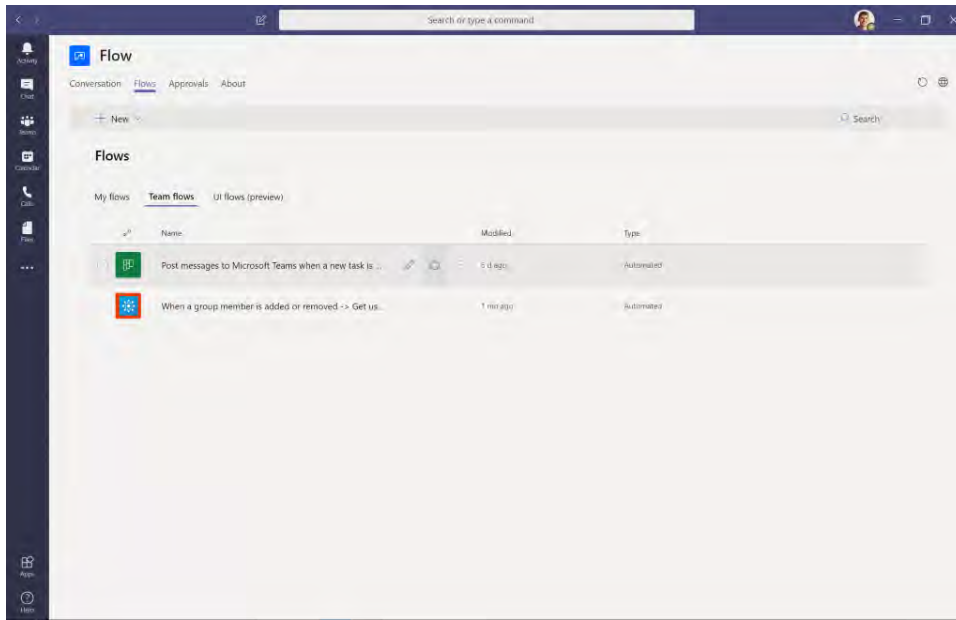


Figure 83 Power Automate in Teams

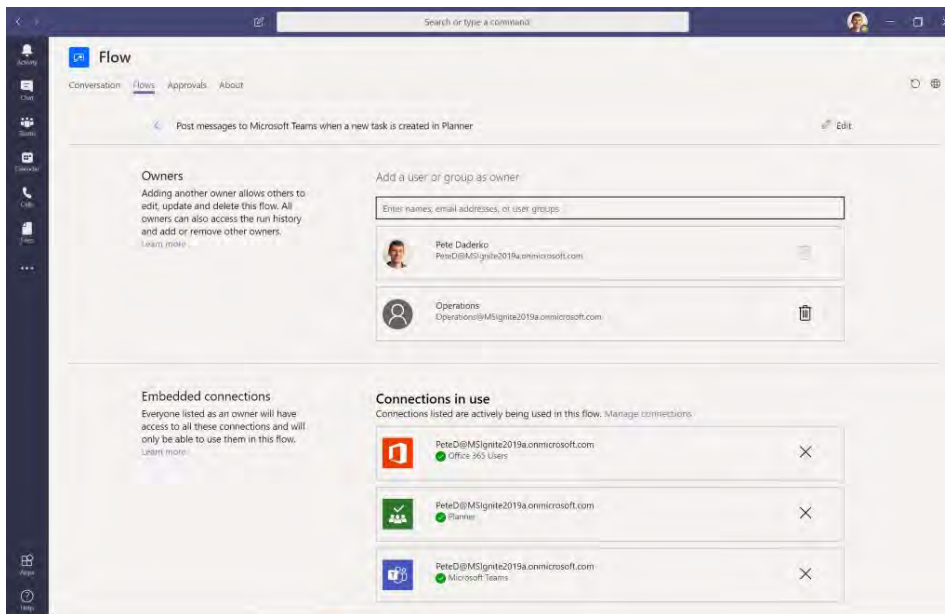


Figure 84 Sharing a flow in Microsoft Teams

Setting app user-access permissions in Teams: Admins can establish granular policies to determine which apps their users are permitted to find and install, including custom apps uploaded for their tenant from Power Apps. In the Teams Admin Center (<https://admin.teams.microsoft.com>), under “Teams apps” select “Permissions policies”, and choose the policy to update (which can be the global policy). All available apps (including Microsoft apps, Third-party apps, and Tenant apps) can be set to “allow all apps”, “block all apps”, or anything in between on an app-by-app basis. Admins can set specific policies

for different user groups if a different set of users need different controls (e.g., the Finance team policies may be different from the Sales team policies).



Figure 85 Teams apps permission policies

Managing app setup in Teams: In the Teams Admin Center, under “Teams apps”, the “Setup policies” feature provides Admins with a great deal of control over the apps in their tenant.

By toggling-on the “Enable custom apps”, users will be permitted to upload custom apps that have been built in Power Apps.

Admins can toggle-on “allow user pinning”, which permits individual users to pin the apps that are most critical to them to the left app bar of Teams.

In addition to user pinning, Admins have control over the appearance of the Teams left app bar. Here, in the “pinned apps” section, Admins can re-order the apps that are pinned to the left bar or even add additional apps to the app bar.

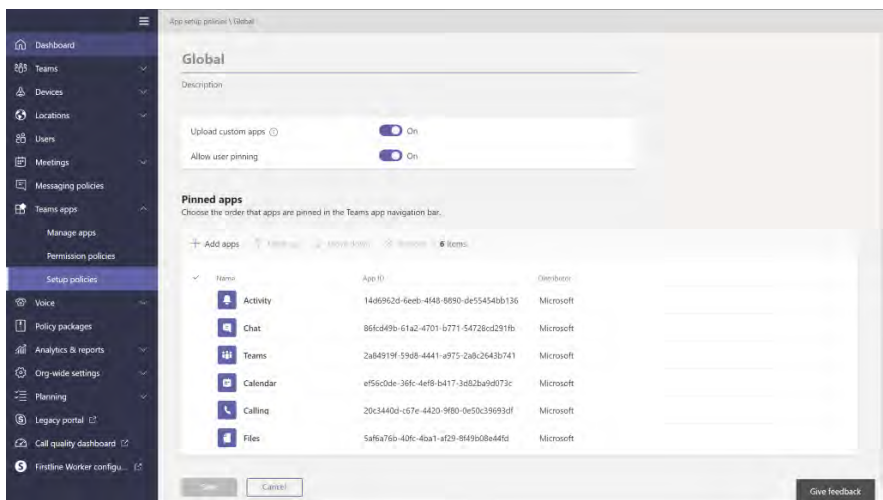


Figure 86 Teams apps Setup policies

Microsoft Teams serves as a platform to surface custom apps and workflows with other users.

Once “Upload custom apps” has been enabled in the Teams Admin Center, canvas apps in Power Apps can easily be added to Teams. These apps can be added either as team apps (apps embedded in a particular chat or channel) or personal apps (those apps that are for individual use and are located on the left app bar of Teams)

Both processes begin in the Power Apps UI by selecting the app to add and clicking the “Add to Teams” button.

Note: these processes apply to individuals with Admin credentials. The experience for non-admin users is slightly different

- In the “Add to Teams” panel, select Download. Power Apps will then generate a Microsoft Teams manifest file using the app description and logo from the app.

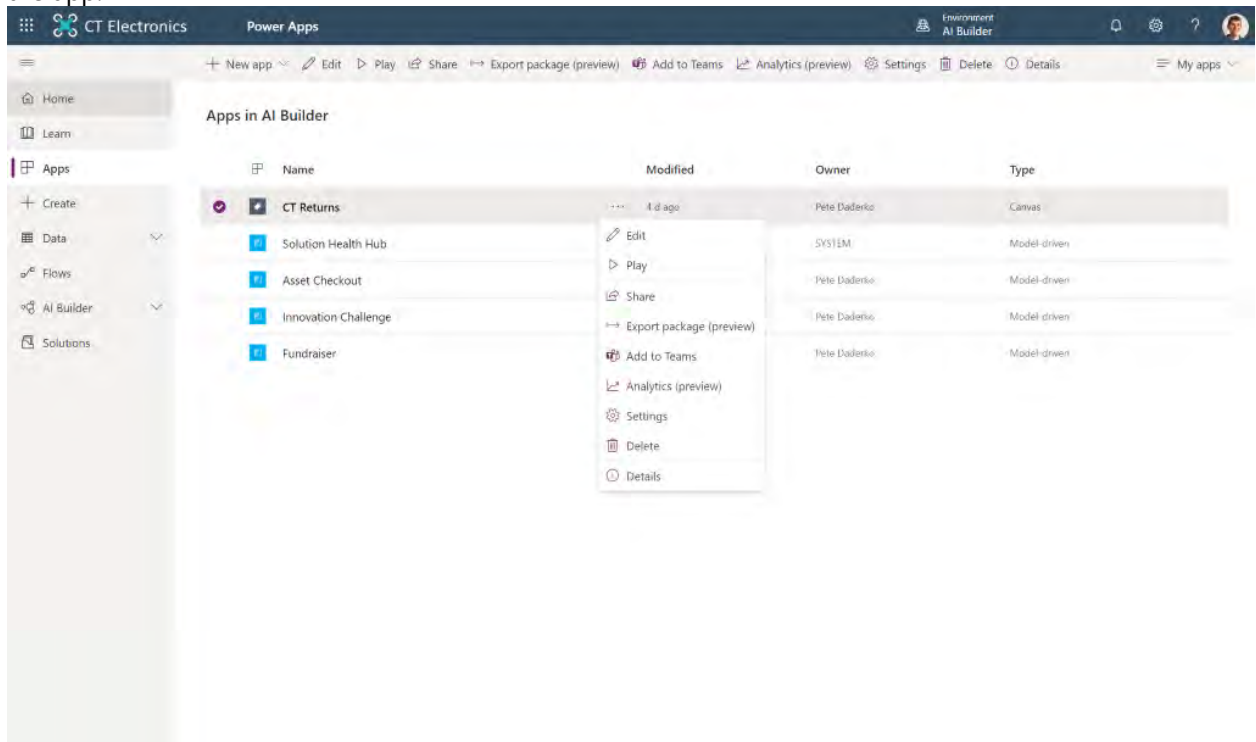


Figure 87 Add to Teams

- Open Microsoft Teams, and select “Apps” in the left navigation, and then select “Upload a custom app”
- Click the “Upload for [your tenant name]” ***note, users without admin-credentials will not see the tenant-level option***
- Select “Add” to add the app to the Teams tenant app catalog.
- To add a canvas app as a personal app:

- Users in the tenant can now access the app from the “Built for [your tenant]” section of the Teams app store to add it as a personal app.
- To add the canvas app as a team app (i.e., pinned to a Teams chat or channel):
- Open the desired channel to add the app to
- Click the “+” button at the top of the channel
- Select the app to add to the channel, and then click “save” once the app details appear

For additional details on administering Power Apps in Teams, see the “Power Apps+Teams” whitepaper (https://aka.ms/powerappsteams_whitepaper)

Flows built with Power Automate can be built, managed, and deployed from within Teams, regardless of whether they are triggered by Teams events or create Teams actions.

Flows that were built outside of Teams will automatically be populated in either the “my flows” list (if flow has not been shared with any other users) or the “team flows” list (for flows that have been shared with other users).

Additionally, users can build and deploy new flows from within Teams using the Power Automate app in Teams. These flows can be built from either templates or blank and have the full capabilities of flows generated in the Power Automate UI.

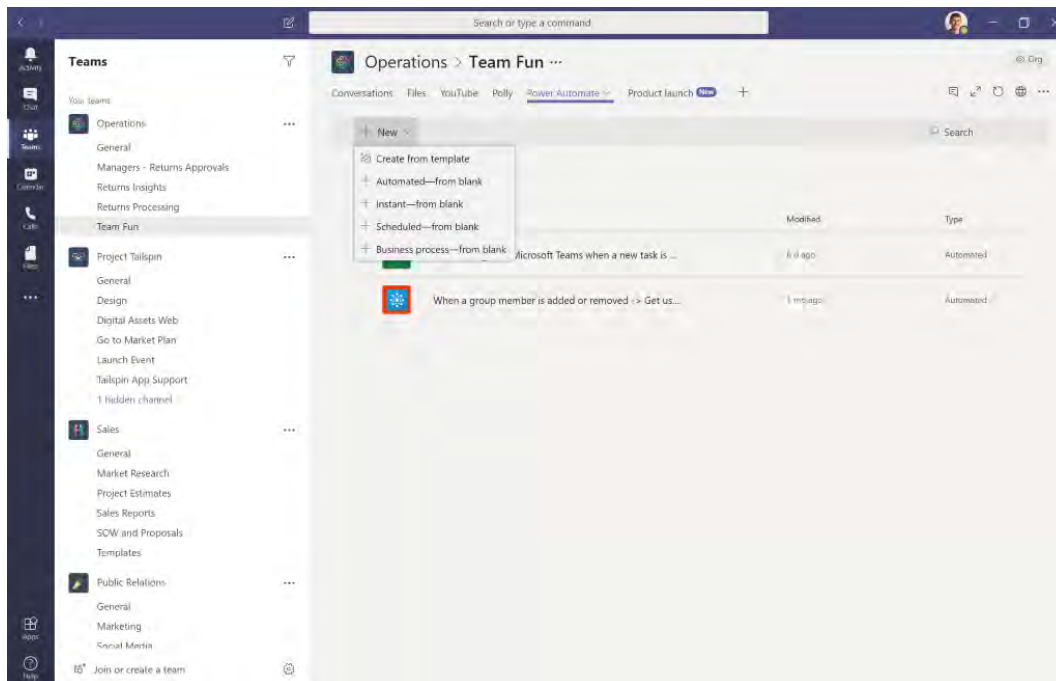


Figure 88 Create flow from Teams

EXPORTING AND IMPORTING APPS

Both types of applications can be exported and then re-imported into other environments, both in the same tenant and in different tenants. Using solutions to package the applications is the recommended approach when there is a CDS database in the source and all target environments. Solutions enable a more complete application life cycle management (ALM) process and are covered in the Application Lifecycle Management chapter of this paper.

EXPORTING AND IMPORTING APPS WITHOUT CDS

Due to backwards compatibility, canvas apps can also be exported standalone for backup or transfer to other environments. The primary use of this feature is moving apps when CDS is not provisioned in the environments and therefore CDS solutions are not an available option. When you export a canvas app into a standalone file, you will choose the action that will be taken in the target environment. You can also choose to add a comment on each resource.

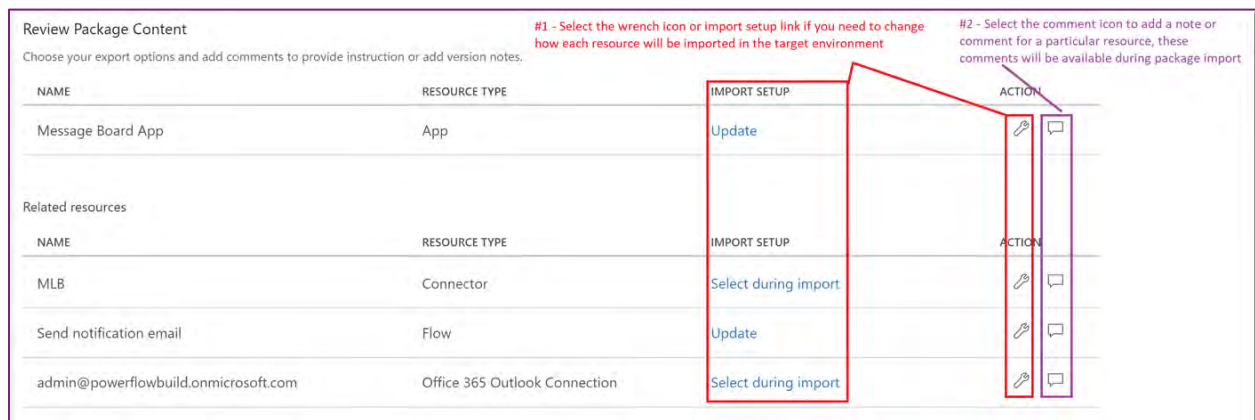


Figure 89: Canvas App Package Content

On import, prior to completion of the import, the related resources will need to be configured to have the proper connections established in the target environment. Custom connectors and CDS customizations will need to be established prior to the import. If the Update action is chosen on import, the new version will be saved as a draft and will need to be "Published" before users will be able to use it. This allows a chance to test the application in the environment without impacting existing users.

It's important to note that the standalone import/export of apps and workflows doesn't offer any of the rich features of solutions. Using the solutions features does require CDS to be configured but it does not require your application to store its data in CDS. By enabling CDS to support solutions use, you gain access to the full solutions application lifecycle (ALM) feature set.

EXPORTING AND IMPORTING FLOWS (POWER AUTOMATE)

Flows can be exported and then re-imported into other environments, both in the same tenant and in different tenants. When CDS is not present, flows export into their own standalone zip file, separate from applications and other CDS components. When CDS is present, flow export functionality is included in the CDS solution framework allowing you to have one solution package that represents all the components in your application.

Flows can also be exported in a Logic App format, allowing conversion of the flow to a Logic App. This capability allows you to move from the flow execution model to the Logic App execution model as well as take advantage of some the more advanced features found in Logic Apps.