

Contents

[Get started](#)

[Start using Azure DevOps](#)

[Start using Azure DevOps](#)

[Overview](#)

[What is Azure DevOps?](#)

[Overview of services](#)

[Azure DevOps Services vs. Azure DevOps Server](#)

[Quickstarts for users](#)

[Code with Git](#)

[Set up continuous integration and delivery](#)

[Plan and track work](#)

[Add and run manual tests](#)

[View permissions](#)

[Quickstarts for admins](#)

[Sign up for Azure DevOps](#)

[Create an organization or project collection](#)

[Manage your project](#)

[Add users to a project or team](#)

[Manage teams and configure team tools](#)

[Tutorials for users](#)

[Set favorites](#)

[Follow work and pull requests](#)

[Get started as a Stakeholder](#)

[Tutorials for admins](#)

[Change individual permissions](#)

[Grant or restrict permissions to select tasks](#)

[Concepts](#)

[Plan your org structure](#)

[Source control](#)

[Clients and tools](#)

[Software development roles](#)

[How-to guides](#)

[Sign in to the web or a client](#)

[Troubleshooting](#)

[Troubleshoot connection](#)

[TF31002: Unable to connect](#)

[Troubleshoot network connections and lists of allowed addresses](#)

[Get support or provide feedback](#)

[Web portal navigation](#)

[Web portal navigation](#)

[Open a service, page, or setting](#)

[Add an artifact or team artifacts](#)

[Use breadcrumbs, selectors, and directories](#)

[Open another project or repo](#)

[Work with favorites](#)

[Work across projects](#)

[Filter basics](#)

[Search your repo, work items, or wiki](#)

[Enable a preview feature](#)

[Reference](#)

[Glossary](#)

[Go mobile](#)

[Keyboard shortcuts](#)

[Search across Azure DevOps](#)

[Search](#)

[About Search](#)

[Search code](#)

[Search work items](#)

[Search packages](#)

[How-to Guides](#)

[Use code search](#)

- [Use work item search](#)
- [Use package search](#)
- [Configure search](#)
- [Troubleshooting](#)
- [FAQs](#)
- [Migrate & import](#)
- [Migrate & Import](#)
- [Migrate data from TFS to Azure DevOps Services](#)
- [Migrate options](#)
- [Import](#)
- [Process templates](#)
- [Post-import](#)
- [Troubleshooting](#)
- [Azure DevOps articles](#)
 - [Add Continuous Security Validation to your CICD Pipeline](#)
 - [Build and Deployment Automation Case Study](#)
 - [Progressively expose your features using feature flags](#)
 - [Progressively expose your releases using deployment rings](#)
- [Reference](#)
 - [Permissions and access \(Security\)](#)
 - [About access levels](#)
 - [Navigate in Team Explorer](#)
 - [FAQs](#)
 - [Service status](#)
 - [Service limits](#)
- [REST API Reference](#)
 - [Search](#)
- [IDE Client Resources](#)
 - [Visual Studio IDE](#)
 - [Visual Studio Code](#)
 - [Visual Studio for Mac](#)
 - [Eclipse](#)

[IntelliJ IDEA](#)

[status & security \(Azure DevOps Services\)](#)

[Service status](#)

[Azure DevOps data protection](#)

[Azure DevOps data location](#)

[Azure DevOps credential storage](#)

[Journey articles](#)

[Agile](#)

[DevOps](#)

[Git at Scale](#)

[Resources](#)

[Configure & customize](#)

[Manage projects](#)

[Marketplace & extensibility](#)

[Search your Wiki](#)

[Code search video](#)

What is Azure DevOps?

8/1/2019 • 5 minutes to read • [Edit Online](#)

[Azure Boards](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

Azure DevOps provides developer services to support teams to plan work, collaborate on code development, and build and deploy applications. Developers can work in the cloud using Azure DevOps Services or on-premises using Azure DevOps Server, formerly named Visual Studio Team Foundation Server (TFS).

Azure DevOps provides an integrated set of features that you can access through your web browser or IDE client. You can acquire one or more of the following services based on your business needs:

- **Azure Repos** provides Git repositories or Team Foundation Version Control (TFVC) for source control of your code
- **Azure Pipelines** provides build and release services to support continuous integration and delivery of your apps
- **Azure Boards** delivers a suite of Agile tools to support planning and tracking work, code defects, and issues using Kanban and Scrum methods
- **Azure Test Plans** provides several tools to test your apps, including manual/exploratory testing and continuous testing
- **Azure Artifacts** allows teams to share Maven, npm, and NuGet packages from public and private sources and integrate package sharing into your CI/CD pipelines
- Collaboration tools that include customizable team dashboards with configurable widgets to share information, progress and trends; built-in wikis for sharing information; configurable notifications and more.

In addition, the Azure DevOps ecosystem also provides support for adding extensions and integrating with other popular services, such as: Campfire, Slack, Trello, UserVoice, and more, and developing your own custom extensions.

For information on the differences between the cloud versus on-premises platforms, see [Azure DevOps Services vs. Azure DevOps Server](#).

Choose Azure DevOps Services

Choose Azure DevOps Services when you want the following results:

- Quick set-up
- Maintenance-free operations
- Easy collaboration across domains
- Elastic scale
- Rock-solid security ([learn more about data protection](#))

You also have access to cloud build and deployment servers, and application insights.

We've made it easy for you to start for free and try out our services.

Sign up for free by creating an organization. Then, either upload your code to share or source control. Begin tracking your work using Scrum, Kanban, or a combination of methods.

You can use all the services included with Azure DevOps, or choose just what you need to complement your existing workflows.

- **Azure Boards** - plan, track, and discuss work across your teams
- **Azure Pipelines** - continuously build, test, and deploy to any platform and cloud
- **Azure Repos** - get unlimited, cloud-hosted private Git repos for your project

Choose Azure DevOps Server

Choose on-premises Azure DevOps Server when your business needs require:

- Your data to stay within your network
- Your work tracking customization requirements are met better with the On-premises XML process model, which supports modification of XML definition files, over those provided with the Inheritance process model.

When you deploy Azure DevOps Server, you can also configure the following servers or integration points:

- **Build server:** supports on-premises and cloud-hosted builds
- **SQL Server and SQL Analysis Server:** supports SQL Server Reports and the ability to create Excel pivot charts based on the cube

Start for free by downloading [Azure DevOps Server Express](#). Then, either upload your code to share or source control. Or, begin tracking your work using Scrum, Kanban, or a combination of methods.

To learn more about managing Azure DevOps Server, see the [Administrative tasks quick reference](#).

Name changes

On September 10, 2018, Microsoft renamed Visual Studio Team Services (VSTS) to Azure DevOps Services. For more information about this change, see [Introducing Azure DevOps](#).

On **November 19, 2018**, the latest version of Team Foundation Server (the on-premises server software for developer collaboration and DevOps services) had a Release Candidate as **Azure DevOps Server 2019 RC 1**.

Visual Studio Team Services is now Azure DevOps Services

Many of the featured services in VSTS are now offered as standalone ones in both Azure DevOps Services and Azure DevOps Server RC 1:

VSTS FEATURE NAME	AZURE DEVOPS SERVICE NAME	DESCRIPTION
Build & release	Azure Pipelines	Continuous integration and continuous delivery (CI/CD) that works with any language, platform, and cloud.
Code	Azure Repos	Unlimited cloud-hosted private Git and Team Foundation Version Control (TFVC) repos for your project.
Work	Azure Boards	Work tracking with Kanban boards, backlogs, team dashboards, and custom reporting.
Test	Azure Test Plans	All-in-one planned and exploratory testing solution.
Packages (extension)	Azure Artifacts	Maven, npm, and NuGet package feeds from public and private sources.

Both Azure DevOps Services and Azure DevOps Server 2019 uses the new navigation user interface, with a vertical sidebar to navigate to the main service areas: **Boards**, **Repos**, **Pipelines**, and more. To learn more, see [Web portal navigation](#).

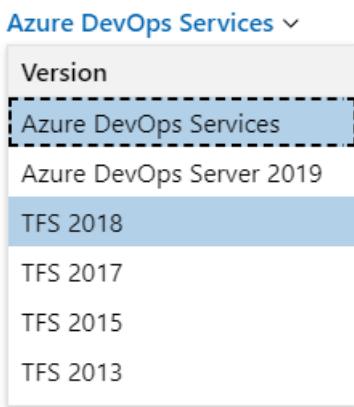
You can acquire each service separately or all together as Azure DevOps Services. If you are already a VSTS subscriber, you have access to all of the services now.

NOTE

You can [disable select services from the user interface](#).

Choose your content version

We support a platform/version selector Toggle the Content version selector dropdown, located above the TOC, to access the content that is specific to your version. The TOC and content page will refresh to show only that content specific to the selected version.



Try this next

[Sign up for Azure DevOps Services](#) or [Install Azure DevOps Server](#)

Related articles

- [A tour of services](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps pricing](#)
- [Azure DevOps release notes](#)
- [Microsoft DevOps blog](#)

Can I still use visualstudio.com to access Azure DevOps Services?

Yes. We've moved to the new `dev.azure.com` domain name as the primary URL for new organizations. (Specifically, it's `https://dev.azure.com/{your organization}/{your project}`.) If you want to change your URL to be based on `dev.azure.com` as the primary, an organization administrator can do so from the organization settings page.

What features and services do I get with Azure DevOps?

9/27/2019 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With Azure DevOps, you gain an integrated set of services and tools to manage your software projects, from planning and development through testing and deployment. Services are delivered through a client/server model. Many of them are delivered through an easy-to-use web interface that you can access from all major browsers. Some services, such as source control, build pipelines, and work tracking, can also be managed through a client.

Access web services through the following areas, as shown in the following image.

The screenshot shows the Azure DevOps Services web interface. At the top, there's a header with a back arrow, forward arrow, refresh button, a lock icon indicating 'Secure' connection, and the URL 'https://'. Below the header is the 'Azure DevOps' logo. The main area has a sidebar on the left with a dark grey header containing the project name 'FabrikamFiber' and a '+' icon. The sidebar lists several items with icons: 'Overview' (blue bar chart), 'Summary' (grey bar chart), 'Dashboards' (grey folder), 'Wiki' (grey document), 'Boards' (green checkmark board), 'Repos' (orange gear), 'Pipelines' (blue pipeline), 'Test Plans' (purple flask), and 'Artifacts' (pink artifact). At the bottom of the sidebar is a 'Logout' button. The main content area has a blue header with the project name 'Fabrikam Fiber' and a dropdown arrow. The header also includes tabs for 'Dashboards', 'Code', 'Work', 'Build & Release', 'Test', and 'Wiki*'. Below the header, there are two tabs: 'Overview' (underlined in blue) and 'Calendar'.

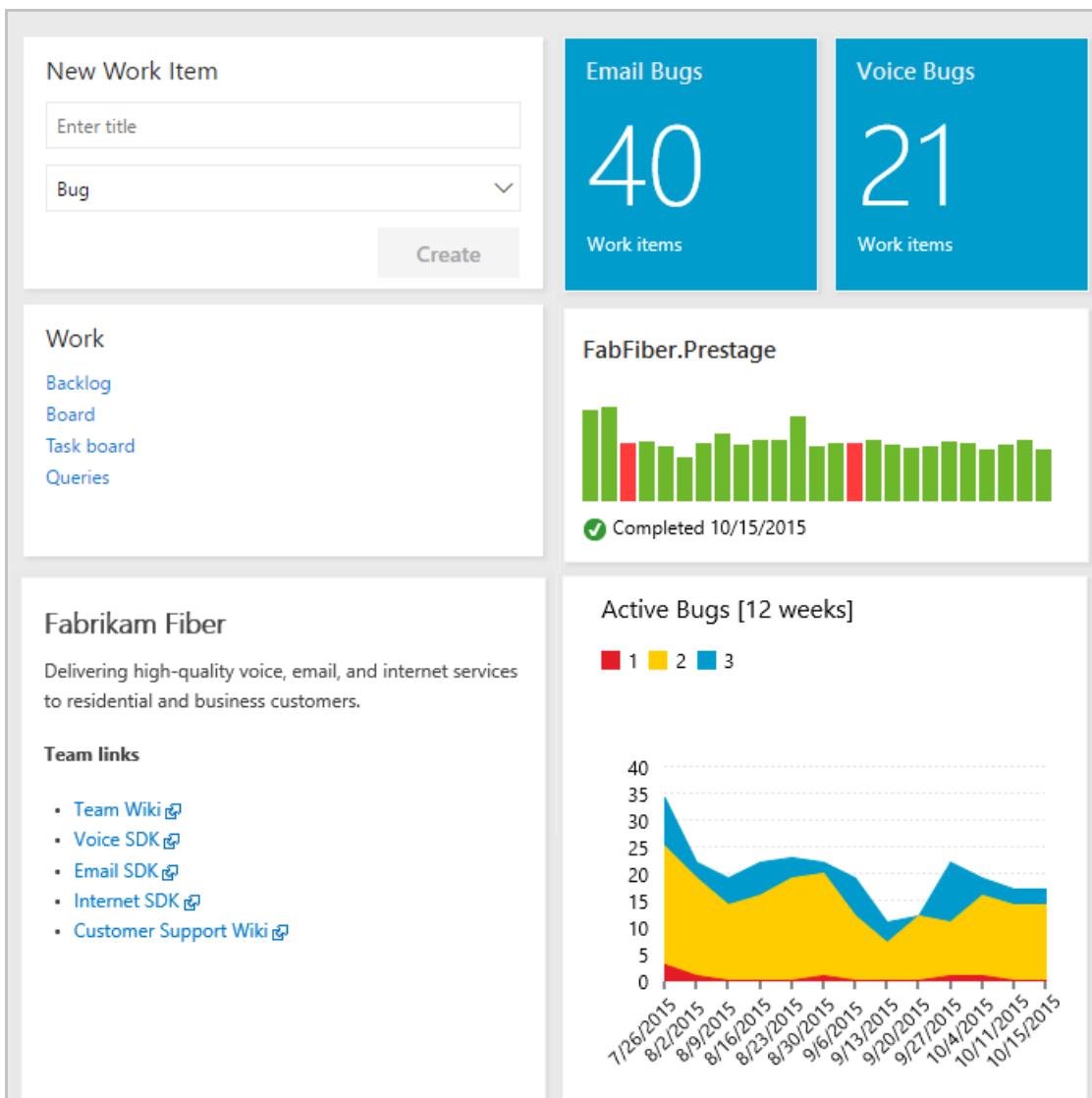
Many of our services are either free for small teams or available through a subscription model or per-use model. You can do a hybrid approach where you use an on-premises deployment to manage your code and work. Then, you purchase cloud build or testing services on an as-needed basis.

For information about client tools, see [Tools](#).

Dashboards

From **Dashboards**, you gain access to user-configurable dashboards.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' team. The left sidebar lists various navigation options: Overview, Summary, Dashboards (which is selected), Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The main content area is titled 'FabrikamFiber Team Overview'. It features a 'Welcome' section with a message: 'Get started using Azure DevOps to make the most of your team dashboard.' Below this are four cards: 'Manage Work' (Add work to your board), 'Collaborate on code' (Add code to your repository), 'Continuously integrate' (Automate your builds), and 'Visualize progress' (Learn how to add charts). A 'Team Members' section shows three icons: 'CC', a person icon, and a plus sign. On the right side, there's a 'Code Tile' section with a 'Configure w...' link and three gear icons.



You can do the following tasks in **Dashboards**:

- Add, configure, and manage dashboards
- Configure widgets that you add to dashboards
- Quickly navigate to different areas of your project

To learn more, see [Dashboards](#).

Source control

Source or version control systems allow developers to collaborate on code and track changes made to the code base. Source control is an essential tool for multi-developer projects.

Our systems support two types of source control: Git (distributed) or Team Foundation Version Control (TFVC), a centralized, client-server system. Both systems enable you to check in files and organize files within folders, branches, and repositories.

With Git, each developer has a copy on their dev machine of the source repository, including all branch and history information. Each developer works directly with their own local repository and changes are shared between repositories as a separate step.

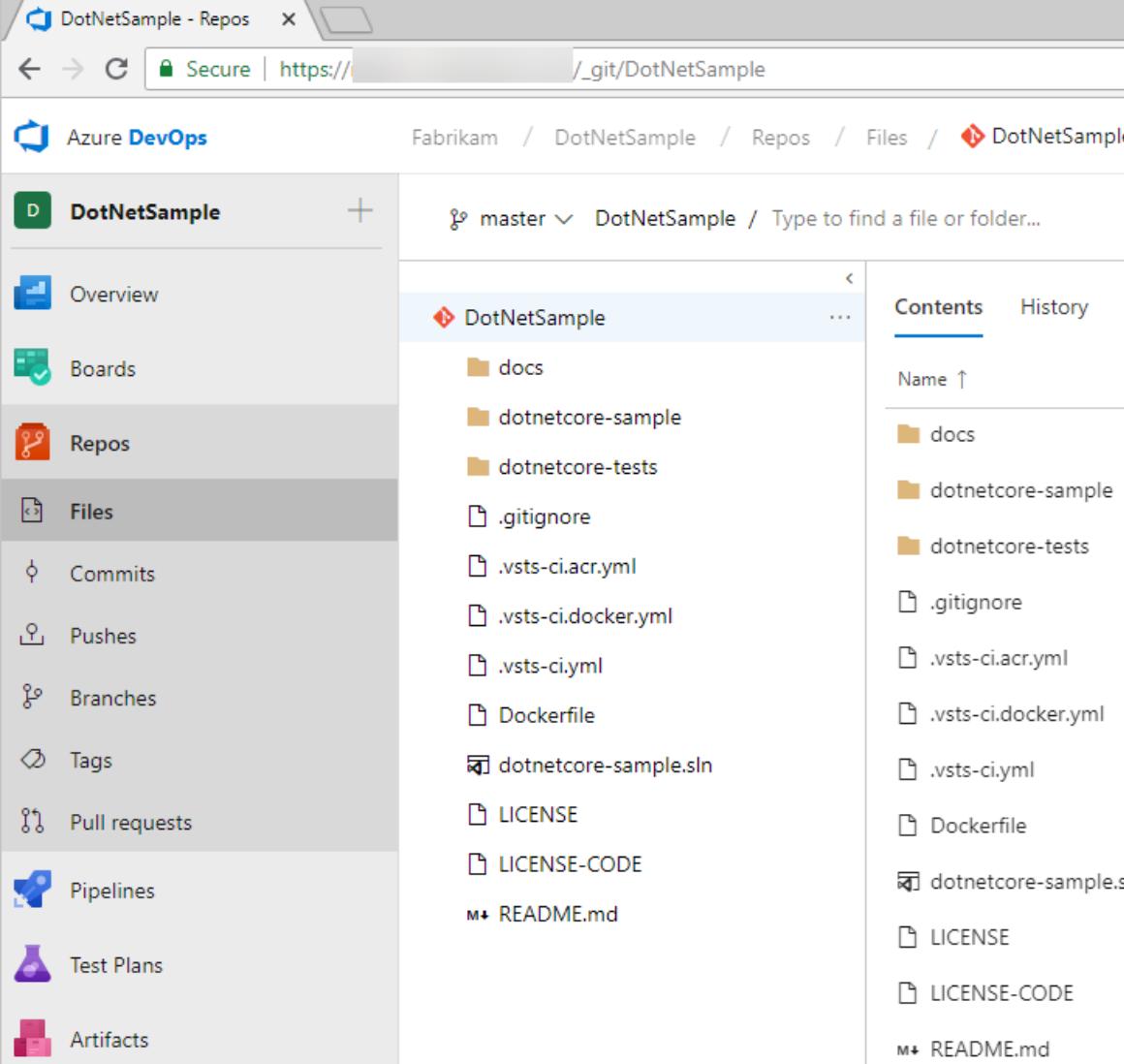
Developers commit each set of changes and do version control operations like history and compare without a network connection. Branches are lightweight. When developers need to switch contexts, they create a private local branch and can switch from one branch to another to pivot among different variations of the codebase. Later, they merge, publish, or dispose of the branch.

NOTE

Git in Azure DevOps is standard Git. You can use Visual Studio with third-party Git services. You can also use third-party Git clients with Azure DevOps Server.

With TFVC, developers have only one version of each file on their dev machines. Historical data is maintained only on the server. Branches are path-based and created on the server.

From **Repos**, you gain access to your source control Git-based or Team Foundation Version Control (TFVC) repositories to support version control of your software projects. These repositories are private.



The screenshot shows the Azure DevOps interface for a repository named "DotNetSample". The left sidebar includes links for Overview, Boards, Repos (selected), Files, Commits, Pushes, Branches, Tags, Pull requests, Pipelines, Test Plans, and Artifacts. The main area displays the contents of the "master" branch of the "DotNetSample" repository. The contents include a "docs" folder, "dotnetcore-sample" and "dotnetcore-tests" folders, a ".gitignore" file, ".vsts-ci.acr.yml", ".vsts-ci.docker.yml", ".vsts-ci.yml", "Dockerfile", "dotnetcore-sample.sln", "LICENSE", "LICENSE-CODE", and "README.md". A "Contents" tab is selected in the ribbon, and a "History" tab is also present.

From **Code**, you gain access to your source control Git-based or TFVC repositories to support version control of your software projects. These repositories are private.

The screenshot shows the Azure DevOps interface for a 'Fabrikam Fiber' repository. The 'Files' tab is selected. The left sidebar shows the repository structure: 'Fabrikam Fiber' / master. The main area displays a table of files under the 'Contents' tab. The columns are: Name, Last change, and Commits. The table includes:

Name	Last change	Commits
page-1.md	10/15/2015	3458a6c7 Added file page-1.md
page-2.md	10/15/2015	01a447ca Added file page-2.md
page-3.md	9/21/2016	68385e28 Added file page-3.md
README.md	5/19/2017	fb9177d8 Merged PR 2: Updated

From Azure Repos for Git, you can do the following tasks:

- Review, download, and edit files, and review the change history for a file
- Review and manage commits that have been pushed
- Review, create, approve, comment on, and complete pull requests
- Add and manage Git tags

To learn more, see the overviews for [Git](#) or [TFVC](#).

Plan and track work

Software development projects require ways to easily share information and track the status of work, tasks, issues, or code defects. In the past, perhaps you used one or more tools. For example, Microsoft Excel, Microsoft Project, a bug tracking system, or a combination of tools. Now, many teams have adopted Agile methods and practices to support planning and development.

Our systems provide several types of work items that you use to track features, requirements, user stories, tasks, bugs, and issues. Each work item is associated with a work item type and a set of fields that can be updated, as progress is made.

For planning purposes, you have access to several types of backlogs and boards to support the main Agile methods—Scrum, Kanban, or Scrumban.

- Product backlog: Used to create and rank stories or requirements.
- Kanban: Used to visualize and manage the flow of work as it moves from beginning, to in-progress, to done.
- Sprint backlogs: Used to plan work to complete during a sprint cycle, a regular two to four-week cadence that teams use when implementing Scrum.
- Task board: Used during daily Scrum meetings to review work that's completed, remaining, or blocked.

Project managers and developers share information by tracking work items on the backlogs and boards. Useful charts and dashboards complete the picture and help teams monitor progress and trends.

From **Boards**, you gain access to Agile tools to support planning and tracking work.

The screenshot shows the Azure DevOps Boards interface for the 'FabrikamFiber' project. On the left, a navigation sidebar lists 'Overview', 'Boards' (selected), 'Work Items', 'Backlogs', 'Sprints', 'Queries', 'Repos', 'Pipelines', 'Test Plans', and 'Artifacts'. The main area displays the 'FabrikamFiber Team' backlog under the 'Stories backlog'. A summary bar at the top right indicates 'Active' items and '3/5'. Below this, two stories are listed:

Story Title	Priority	Status
Technician can report busy/late on Windows Phone	3	Pending
Technician can see service tickets on Windows Phone	1	In Progress

A 'New item' button is available to add more stories.

From **Work**, you gain access to Agile tools to support planning and tracking work.

The screenshot shows the Azure DevOps Backlogs interface for the 'Fabrikam Fiber' project. The left sidebar shows 'Epics', 'Features', and 'Stories' (selected). Under 'Stories', there are sections for 'Past' (Sprint 1, Sprint 2) and 'Current' (Sprint 3, Sprint 4, Sprint 5). The main area shows the 'Stories' backlog with the following details:

Type	Title	Order	State	Story Points	Title
User Story	Add an information form	1	New	5	Add an information form
User Story	Welcome back page	2	New	3	Welcome back page
User Story	Interim save on long forms	3	New	8	Interim save on long forms
User Story	Secure Sign-in	4	Active	5	Secure Sign-in
User Story	Canadian addresses don't display	5	Active	5	Canadian addresses don't display

Specifically, you can do the following tasks:

- Add and update work items
- Define work item queries, and create status and trend charts based on those queries
- Manage your product backlog

- Plan sprints by using sprint backlogs
- Review sprint tasks and update tasks through the task boards
- Visualize the workflow and update the status by using Kanban boards
- Manage portfolios by grouping stories under features and grouping features under epics

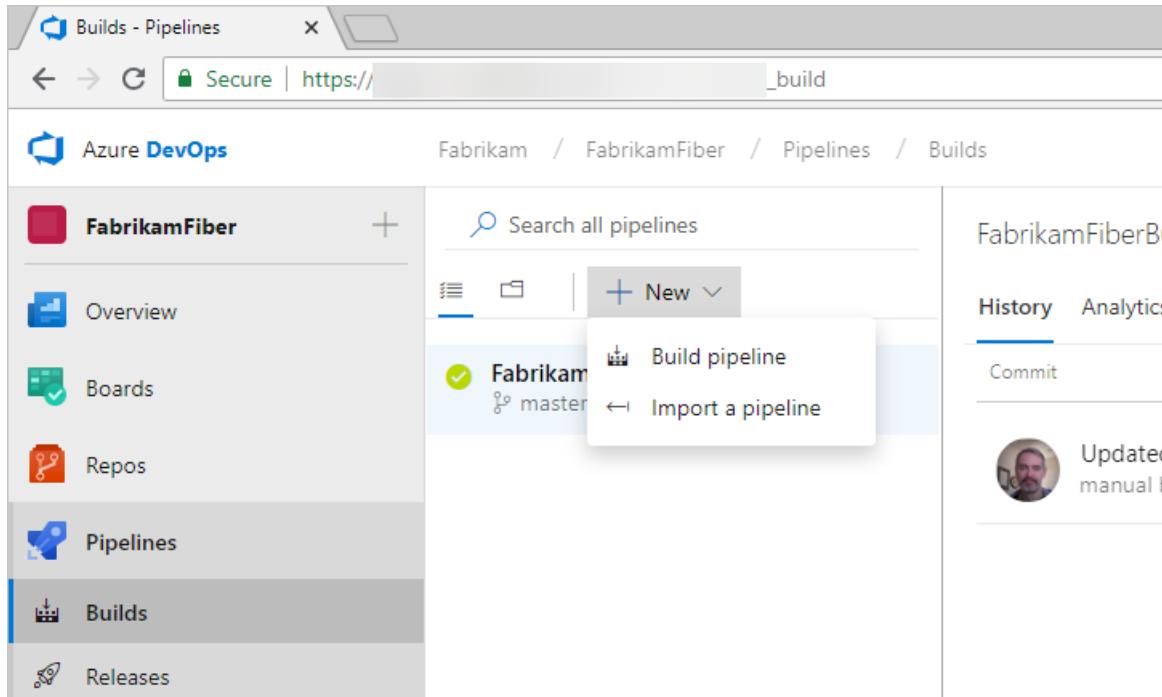
See [Backlogs, boards, and plans](#) for an overview of each.

Continuous integration and deployment

The rapid and reliable release of software comes from automating as many processes as possible. Our systems support build, test, and release automation.

- You can define builds to automatically run whenever a team member checks in code changes.
- Your build pipelines can include instructions to run tests after the build runs.
- Release pipelines support managing deployment of your software builds to staging or production environments.

Azure Pipelines provides an integrated set of features to support building and deploying your applications.



Azure Pipelines provides an integrated set of features to support building and deploying your applications.

The screenshot shows the Microsoft DevOps interface for managing build definitions. At the top, there's a navigation bar with links for Fabrikam Fiber, Dashboards, Code, Work, Build & Release, Test, and more. Below that is a secondary navigation bar with tabs for Builds, Releases, Library, Task Groups, and Deployment Groups*. The main area is titled "Build Definitions" and includes a search bar for "Build ID or build number" and buttons for "+ New" and "+ Import". There are also filters for "Mine", "All Definitions", "Queued", and "XAML". A section titled "My favorites" lists four recent builds:

Build	Status	Triggered by	7-day pass rate
Content.VS Build : #Content.VS Build_20160609.1 requested a year ago	passing	Updated the overview s... 80496e4 in 8 users/...	0% →
Content.VS.PR : #Content.VS.PR_20161019.14 requested 10 months ago	passing	Merge pull request 152... 2be71b1 in 152638	0% →
MSDN.GatedCheck.ALM-master : #20170313.2 requested 5 months ago	passing	Merge pull request 194... 8f7955d in 194899	0% →
MSDN.GatedCheck.VS-master : #20160725.1 requested a year ago	passing	Merge pull request 126... 2d56c79 in 126293	0% →

Use pipelines to implement continuous integration and continuous delivery.

- **Build automation:** Define the steps to take during build and the triggers that start a build.
- **Release management:** Supports a rapid release cadence and management of simultaneous releases. You can configure release pipelines that represent your environments from development to production. Run automation to deploy your app to each environment. Add approvers to confirm that the app has been successfully deployed in an environment. Create your release manually or automatically from a build. Then track your releases as they're deployed to various environments.

To learn more, see [Continuous integration on any platform](#).

Manual and exploratory testing

Test features support manual and exploratory testing, and continuous testing.

Test Plans supports creating and managing manual tests.

The screenshot shows the Microsoft Test Plans interface. At the top, it says "Test Plans > FabrikamFiber". The main area displays a "Test suite: FabrikamFiber (Suite ID: 367)". Below that are tabs for "Tests" and "Charts". On the left, there's a sidebar with icons for creating new suites and a list showing "FabrikamFiber (2)" with a "New suite" option. The "Tests" tab is selected, showing a table with two rows:

Outcome	Order	ID	Title
Active	1	368	Fabrikam Test
Active	2	369	Test sign in flow

Test supports creating and managing manual tests.

The screenshot shows the Microsoft DevOps Test Plans interface. At the top, there's a navigation bar with links for Dashboards, Code, Work, Build & Release, Test, Wiki*, and a gear icon. Below the navigation bar, there's a sub-navigation menu with links for Test Plans, Parameters, Configurations, Runs, Machines, and Load test. The main area displays a "Test suite: 379 : Phone sign in (Suite ID: 477)" grid. The grid has columns for Tests, Charts, Outcome All, Tester All, and Configuration All. It includes buttons for New, Add existing, Run, and other actions. The data grid shows two rows of test cases:

Outcome	Order	ID	Title	Configuration
Active	1	474	[redacted]	Windows 8
Active	2	478	[redacted]	Windows 8

With test features, you gain access to the following features:

- Customization of workflows with test plan, test suite, and test case work items
- End-to-end traceability from requirements to test cases and bugs with requirement-based test suites
- Criteria-based test selection with query-based test suites
- Excel-like interface with the grid for easy creation of test cases
- Reusable test steps and test data with shared steps and shared parameters
- Sharable test plans, test suites, and test cases for reviewing with Stakeholders
- Browser-based test execution on any platform
- Real-time charts for tracking test activity

To learn more, see [Testing overview](#).

Collaboration services

The following services work across the previously mentioned services to support:

- Team dashboards
- Project wiki
- Discussion within work item forms
- Linking of work items, commits, pull requests, and other artifacts to support traceability
- Alerts and change notifications managed per user, team, project, or organization
- Ability to request and manage feedback
- Analytics service, analytic views, and Power BI reporting
- Dashboards
- Project wiki
- Discussion within work item forms
- Linking of work items, commits, pull requests, and other artifacts to support traceability
- Alerts and change notifications managed per user, team, project, or project collection
- Ability to request and manage feedback
- SQL Server Reporting
- Dashboards
- Discussion within work item forms
- Linking of work items, commits, pull requests and other artifacts to support traceability
- Alerts and change notifications managed per user, team, project, or project collection
- Ability to request and manage feedback
- Team (chat) rooms

- SQL Server Reporting

NOTE

Team rooms are deprecated for TFS 2017.2. Instead, we recommend that you [use service hooks to integrate with Microsoft Teams](#).

- Dashboards
- Linking of work items, commits, pull requests, and other artifacts to support traceability
- Alerts and change notifications managed per user or for teams
- Ability to request and manage feedback
- Team (chat) rooms
- SQL Server Reporting
- Team home page
- Linking of work items, commits, pull requests, and other artifacts to support traceability
- Alerts and change notifications managed per user or for teams
- Ability to request and manage feedback
- Team (chat) rooms
- SQL Server Reporting

Service hooks

Service hooks enable you to complete tasks on other services when events happen within your project hosted on Azure DevOps. For example, you can send a push notification to your team's mobile devices when a build fails. You can also use service hooks in custom apps and services as a more efficient way to drive activities in your projects.

The following services are available as the target of service hooks. To learn about other apps and services that integrate with Azure DevOps, visit the [Visual Studio Marketplace](#), Azure DevOps tab.

For the latest set of supported services, see [Integrate with service hooks](#).

Cloud-hosted services based on usage

The following services support your DevOps operations:

- Cloud-based, Microsoft-hosted build and deployment agents
- On-premises self-hosted agents to support build and deployment

To learn more, see [Pricing](#).

Azure cloud-hosted services

Azure provides cloud-hosted services to support application development and deployment. You can make use of these services solely or in combination with Azure DevOps.

To browse the directory of integrated services, features, and bundled suites, see [Azure products](#).

For continuous delivery to Azure from Azure DevOps Services, see [Automatically build and deploy to Azure web apps or cloud services](#).

Administrative services

There are features and tasks associated with administering a collaborative software development environment. You complete most of these tasks through the web portal. To learn more, see [About user, team, project, and organization-level settings](#).

The screenshot shows the Azure DevOps Project Settings Overview page for the 'FabrikamFiberTest' project. The left sidebar lists project navigation options: Overview, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The 'Project settings' option is highlighted with a red box. The main content area displays 'Project details' with fields for Name (FabrikamFiberTest) and Description (Fabrikam Fiber test project). It also shows the Process (Scrum), Visibility (Private), and a Save button. Below this, the 'Azure DevOps services' section is shown with icons and links for Boards, Repos, Pipelines, and Packages.

Project Settings > Overview

Project details

Name: FabrikamFiberTest

Description: Fabrikam Fiber test project

Process: Scrum

Visibility: Private

Save

Azure DevOps services

- Boards** Flexible agile planning
- Repos** Repos, pull requests, ad
- Pipelines** Build, manage, and scale
- Packages**

Project settings

S Fabrikam Fiber Dashboards Code Work Build & Release Test Wiki* | 

Overview Work Security Version Control Policies Agent queues Notifications Service Hooks

Project profile

 Teams

New team | 

Team Name ↑	Members	Description
 Customer Service	7	
 Fabrikam Fiber Team	7	The default project team.
 Management team	1	
 Phone	1	
 Voice	1	
 Web	2	

Name
Fabrikam Fiber

Process
Scrum

Description
Web, voice, and phone apps

Related articles

- [Understand differences between Azure DevOps Services and Azure DevOps Server](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps pricing](#)
- [Azure DevOps data protection overview](#)

Azure DevOps Services vs. Azure DevOps Server

9/4/2019 • 10 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Azure DevOps Services and Azure DevOps Server were formerly named Visual Studio Team Services (VSTS) and Team Foundation Server (TFS). Both offerings provide an integrated, collaborative environment that supports Git, continuous integration, and Agile tools for planning and tracking work.

Azure DevOps Services is the **cloud offering** that provides a scalable, reliable, and globally available hosted service. It's backed by a 99.9% SLA, monitored by our 24/7 operations team, and available in local data centers around the world.

Azure DevOps Server is the **on-premises offering** that's built on a SQL Server back end. Companies usually choose on-premises when they need their data to stay within their network or when they want access to SQL Server reporting services that integrate with Azure DevOps data and tools.

Although both offerings provide the same [essential services](#), compared with Azure DevOps Server, Azure DevOps Services offers the following added benefits:

- Simplified server management.
- Immediate access to the latest and greatest features
- Improved connectivity with remote sites.
- A transition from capital expenditures (servers and the like) to operational expenditures (subscriptions).

To determine which offering—cloud or on-premises—meets your needs, consider the following key differences.

Fundamental differences between Azure DevOps Services and Azure DevOps Server

When you're choosing which platform you want, or if you're considering a move from on-premises to the cloud, consider the following areas:

- [Scope and scale data](#)
- [Authentication](#)
- [Users and groups](#)
- [Manage user access](#)
- [Security and data protection](#)

Differences in specific feature areas

Although Azure DevOps Services is a hosted version of Azure DevOps Server, there are some differences between features. Some Azure DevOps Server features aren't supported in Azure DevOps Services. For example, Azure DevOps Services doesn't support integration with SQL Server Analysis Services to support reporting.

Two of the following additional areas differ in their support:

- [Process customization](#)
- [Reporting](#)

Are you on Azure DevOps Server and considering moving? Read [Migration options](#) to understand your options.

Scope and scale data

Azure DevOps Services scales by using organizations and projects

Azure DevOps Services differs slightly from Azure DevOps Server. There are currently only two options for scoping and scaling data: organizations and projects. Organizations in Azure DevOps Services get their own URLs (for example, <https://dev.azure.com/fabrikamfiber>), and they always have exactly one project collection.

Organizations can have many projects within a collection.

We recommend that you create organizations in Azure DevOps Services wherever you would create collections in Azure DevOps Server. The following scenarios apply:

- You can purchase Azure DevOps Services users per organization - Paid users can access only the organization in which the payment is made. If you have users who need access to many organizations, Visual Studio subscriptions can be an attractive option. Visual Studio subscribers can be added to any number of organizations at no charge. We're also considering other ways to make access available to many organizations that are grouped into a single organization.
- You currently have to administer organizations one at a time. This process can be cumbersome when you have many organizations.

Learn more: [Plan your organizational structure in Azure DevOps](#).

Azure DevOps Server scales by using deployments, project collections, and projects

Azure DevOps Server offers the following three options for scoping and scaling data: deployments, project collections, and projects. In the simplest case, deployments are just servers.

Deployments can be more complicated, however, which could include:

- Two-server deployment where SQL is split out on a separate machine
- High-availability farms with lots of servers

Project collections serve as containers for security and administration, and physical database boundaries. They're also used to group related projects.

Finally, projects are used to encapsulate the assets of individual software projects, including source code, work items, and so on.

To learn more, see [Plan your Azure DevOps organization structure](#).

Authentication

With Azure DevOps Services, you connect over the public internet (for example, <https://contoso.visualstudio.com>). You either authenticate with [Microsoft account](#) credentials or with [Azure AD](#) credentials, depending on your organization setup. You can also set up Azure AD to require features such as multi-factor-authentication, IP address restrictions, and so on.

We recommend that you configure your organizations to use Azure AD rather than Microsoft accounts. This method provides a better experience in many scenarios and more options for enhanced security.

Learn more: [Access Azure DevOps Services with Azure Active Directory](#).

With Azure DevOps Server, you connect to an intranet server (for example, <https://tfs.corp.contoso.com:8080/tfs>). You authenticate with Windows Authentication and your Active Directory (AD) domain credentials. This process is transparent and you never see any kind of sign in experience.

Manage users and groups

In Azure DevOps Services, you can use a similar mechanism to [provide access to groups of users](#). You can add

Azure AD groups to Azure DevOps Services groups. If you use Microsoft Accounts instead of Azure AD, you have to [add users](#) one at a time.

In Azure DevOps Server, you provide users access to deployments by adding Active Directory (AD) groups to various Azure DevOps groups (for example, the Contributors group for an individual project). The AD group memberships are kept in sync. As users are added and removed in AD, they also gain and lose access to Azure DevOps Server.

Manage user access

In both Azure DevOps Services and Azure DevOps Server, you manage access to features by assigning users to an [access level](#). All users must be assigned to a single access level. In both the cloud and on-premises offerings, you can give free access to work item features to an unlimited number of Stakeholders. Also, an unlimited number of Visual Studio subscribers can have access to all Basic features at no additional charge. You pay only for other users who need access.

In Azure DevOps Services, you must [assign an access level](#) to each user in your organization. Azure DevOps Services validates Visual Studio subscribers as they sign in. You can assign Basic access for free to five users without Visual Studio subscriptions.

To give Basic access or higher to more users, [set up billing](#) for your organization and [pay for more users](#). Otherwise, all other users get Stakeholder access.

If you use Azure AD groups to give access to groups of users, access levels are automatically assigned at first sign-in. For organizations that are configured to use Microsoft accounts for signing in, you must assign access levels to each user explicitly.

In Azure DevOps Server, all use is on the honor system. To set access levels for users based on their licenses, specify their [access levels](#) on the administration page. For example, assign unlicensed users Stakeholder access only.

Users with an Azure DevOps Server/TFS Client Access License (CAL) can have Basic access. Visual Studio subscribers can have either Basic or Advanced access, depending on their subscriptions. Azure DevOps Server doesn't attempt to verify these licenses or enforce compliance.

Security and data protection

Many entities want to know more about data protection when they consider moving to the cloud. We're committed to ensuring that Azure DevOps Services projects stay safe and secure. We have technical features and business processes in place to deliver on this commitment. You can also take steps to secure your data. Learn more in our [Data Protection overview](#).

Process customization

You customize the work-tracking experience in two different ways, depending on the supported process model:

- For Azure DevOps Services, you use the **Inheritance** process model, which supports WYSIWYG customization.
- For Azure DevOps Server, you can choose the **Inheritance** process model or the **On-premises XML** process model, which supports customization through import or export of XML definition files for work-tracking objects.
- For Azure DevOps Server 2018 and earlier versions, you only have access to the **On-premises XML** process model.

Although the **On-premises XML** process model option is powerful, it can cause various issues. The main issue is that processes for existing projects aren't automatically updated.

For example, Azure DevOps Server 2013 introduced several new features that depended on new work-item types and other process template changes. When you upgrade from 2012 to 2013, each project collection gets new versions of each of the "in the box" process templates that include these changes. However, these changes aren't automatically incorporated into existing projects. Instead, after you finish upgrading, you have to include the changes in each project by using the [Configure features](#) wizard or a more manual process.

To help you avoid these issues in Azure DevOps Services, custom process templates and the **witadmin.exe** tool have always been disabled. This approach has enabled us to automatically update all projects with each Azure DevOps Services upgrade. Meanwhile, the product team is working hard to make customizing processes possible in ways that we can support easily and continuously. We recently introduced the first of these changes and more changes are on the way.

With the new process-customization capability, you can make changes directly within the web user interface (UI). If you want to customize your processes programmatically, you can do so through REST endpoints. When you customize projects this way, they're automatically updated when we release new versions of their base processes with Azure DevOps Services upgrades.

To learn more, see [Customize your work-tracking experience](#).

Reporting

Azure DevOps Services and Azure DevOps Server offer a many tools that give you insight into the progress and quality of your software projects. Included are the following tools:

- [Dashboards](#) and lightweight [charts](#) that are available in both the cloud and on-premises platforms. These tools are easy to set up and use.

Azure DevOps Services and Azure DevOps Server 2019 also provide access to the following services:

- [The Analytics service](#) and [Analytics widgets](#). The Analytics service is optimized for fast read-access and server-based aggregations.
- [Microsoft Power BI integration](#), which supports getting Analytics data into Power BI reports and provides a combination of simplicity and power.
- [OData support](#), which allows you to directly query the Analytics service from a supported browser, and then use the returned JSON data as you want. You can generate queries that span many projects or your entire organization.

To learn more about the Analytics service and future releases, see our [Reporting roadmap](#).

[SQL Server Reporting Services \(SSRS\) reports](#) are available from Azure DevOps Server or TFS when configured with SQL Server Analysis Services.

Related articles

- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps Services - pricing](#)
- [Azure DevOps Server - pricing](#)

Quickstart: Code with Git

8/1/2019 • 3 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this quickstart, you learn how to share your code with others. After you create a new organization and project in Azure DevOps, you can begin coding with Git.

To work with a Git repo, you clone it to your computer. Cloning a repo creates a complete local copy of the repo for you to work with. Cloning also downloads all [commits](#) and [branches](#) in the repo, and sets up a named relationship with the repo on the server. Use this relationship to interact with the existing repo, pushing and pulling changes to share code with your team.

Install Git command-line tools

1. Install one of the following Git command-line tools:

- To install Git for Windows, including Git Credential Manager, see [Install the Git Credential Manager - Windows](#).
- To install on macOS or Linux, check out the [Installing Git](#) chapter in the open-source *Pro Git* book. For macOS and Linux, we recommend [configuring SSH authentication](#)

Clone the repo to your computer

1. From your web browser, open the project for your organization, and select **Repos**. If you don't have a project, [create one now](#).

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' project. The left sidebar has a 'Repos' section highlighted with a red box. A dropdown menu titled 'Clone to your computer' is open, listing various cloning options like 'Files', 'Commits', and 'Branches'. The URL 'https://dev.azure.com/FabrikamFiber/_git/FabrikamFiber' is visible in the address bar.

2. Select **Clone** in the upper-right corner of the Code window, and copy the URL.

The screenshot shows the 'Files' page for the 'FabrikamFiber' project. The left sidebar has a 'Repos' section selected. In the main area, there's a 'Clone to your computer' section with an 'HTTPS' tab selected, showing the URL 'https://dev.azure.com/FabrikamFiber/_git/FabrikamFiber'. A red box highlights the copy icon next to the URL.

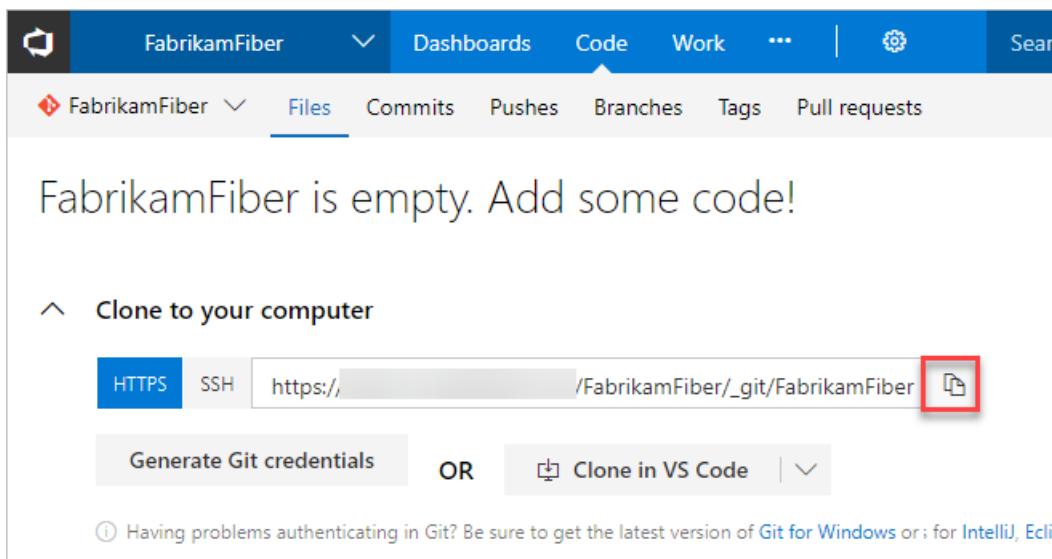
3. Open the Git command window (Git Bash on Git for Windows). Go to the folder where you want the code from the repo stored on your computer, and run `git clone`, followed by the path copied from **Clone URL** in the previous step. See the following example:

```
git clone https://contoso-ltd.visualstudio.com/MyFirstProject/_git/contoso-demo
```

A copy of the code is downloaded in Git, including all **commits** and **branches** from the repo. This copy is put into a new folder for you to work with.

Keep this command window open (you'll use it in the following steps).

1. From your web browser, open the project for your organization, and select **Code**. If you don't have a project, [create one now](#).
2. Select **Clone** in the upper-right corner of the Code window, and copy the URL.



The screenshot shows the Visual Studio Code interface with the 'Code' tab selected. In the center, it says 'FabrikamFiber is empty. Add some code!'. Below this, under 'Clone to your computer', there are two options: 'HTTPS' (selected) and 'SSH'. The 'HTTPS' URL is displayed as 'https:// /FabrikamFiber/_git/FabrikamFiber'. To the right of the URL is a copy icon, which is highlighted with a red box. Below the URLs, there's a note: 'Having problems authenticating in Git? Be sure to get the latest version of [Git for Windows](#) or [for IntelliJ, Eclipse, and other IDEs](#)'.

3. Open the Git command window (Git Bash on Git for Windows). Go to the folder where you want the code from the repo stored on your computer, and run `git clone`, followed by the path copied from **Clone URL** in the previous step. See the following example:

```
git clone https://contoso-ltd.visualstudio.com/MyFirstProject/_git/contoso->demo
```

A copy of the code downloads into a new folder in Git. The code includes all of your [commits](#) and [branches](#) from the repo.

Keep the command window open (use it in the following steps).

Work with the code

In the following steps, we make a change to the files on your computer, commit the changes locally, and push the commit to the repo stored on the server. We can then view the changes.

1. Browse to the folder on your computer where you cloned the repo, open the `README.md` file in your editor of choice, and make some changes. Then save and close the file.
2. In the Git command window, go to the `contoso-demo` directory by entering the following command:

```
cd contoso-demo
```

3. Commit your changes by entering the following commands in the Git command window:

```
git add .
git commit -m "My first commit"
```

The `git add .` command stages any new or changed files, and `git commit -m` creates a commit with the specified commit message.

4. Push your changes to the Git repo on the server. Enter the following command into the Git command window:

```
git push
```

View history

1. Switch back to the web portal, and select **History** from the Code page to view your new commit.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' repository. The 'History' tab is selected. A single commit is listed:

- Wednesday, August 8, 2018 1 changeset**
Added file README
Jamal Hartnett created #1528, 8/8/2018

2. Switch to the **Files** tab, and select the README file to view your changes.

The screenshot shows the 'Files' tab for the 'README' file. The commit message is displayed:

① Checked in changeset ⌂ 1529: Updated README

Contents History Compare Annotate

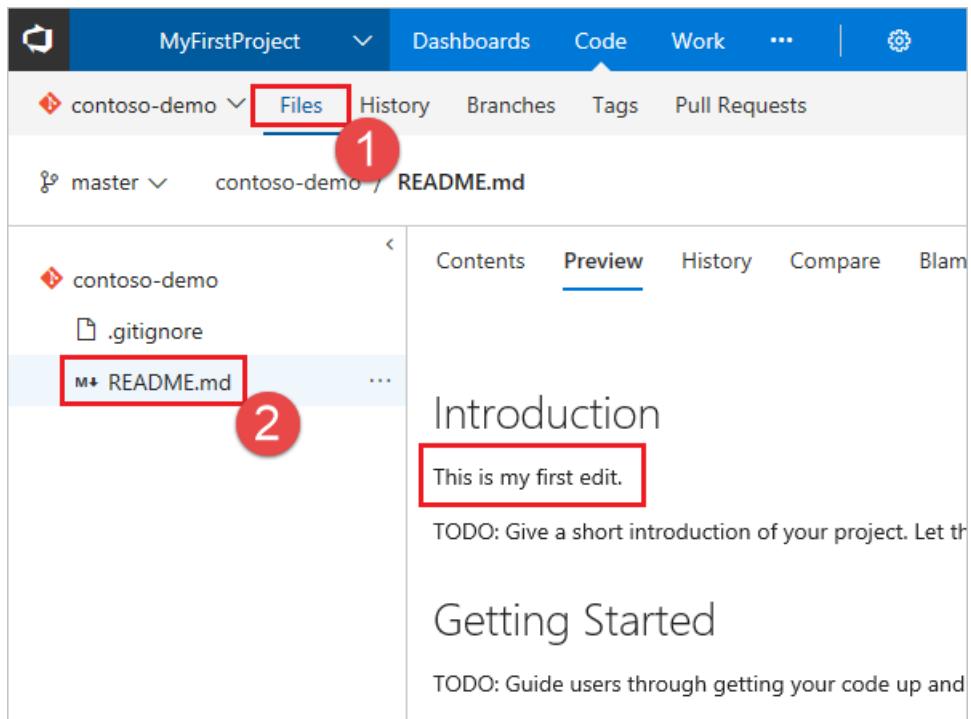
1 My first edit to README file.

1. Switch back to the web portal, and select **History** from the **Code** tab to view your new commit. Two commits appear: the first commit, where the README and .gitignore were added upon repo creation, and the commit you just made.

The screenshot shows the 'Code' tab with the 'History' sub-tab selected. The commit history is shown in a table:

Graph	Commit	Message
4b38f92b	My first commit	
fefb3a74	Added README.md, .gitignore (VisualStudio) files	

2. Switch to the **Files** tab, and select the README file to view your changes.



Next steps

[Set up continuous integration & delivery](#)

2 minutes to read

Plan and track work

9/27/2019 • 15 minutes to read • [Edit Online](#)

Azure Boards | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You track your work by creating work items. This article walks you through creating issues and tasks using a Kanban board for the Basic process, or creating user stories and tasks using for the Agile process.

NOTE

The Basic process is currently only available from Azure DevOps Services. For on-premises deployments, choose the Agile process.

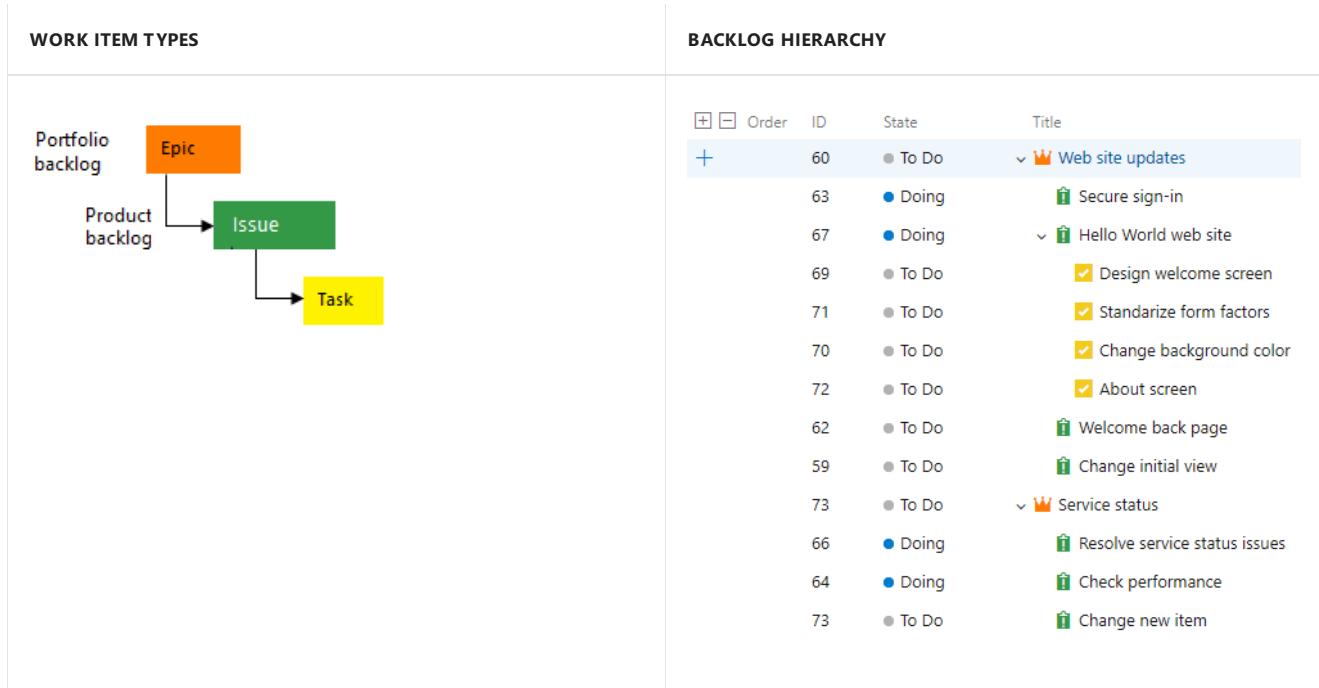
Choose either the **Basic process** or **Agile process** for guidance depending on what process was selected for your project.

NOTE

The Basic process is available when you add a project to Azure DevOps Services or [Azure DevOps Server 2019 Update 1](#). For earlier on-premises deployments, choose Agile, Scrum, or CMMI process.

- [Basic process](#)
- [Agile process](#)

The Basic process provides three work item types—epics, issues, and tasks—to plan and track work. We recommend you start by adding issues to track your user stories, bugs, or feature items. If you need to group them into a hierarchy, you can define epics. If you want to track additional details of work, you can add tasks to an issue.



Within each work item form, you can describe the work to be done, assign work to project contributors, track status, and collaborate with others through the Discussion section.

Here we show how to add issues and child tasks from the web portal and add details to those work items .

Open the Kanban board

- Basic process
- Agile process

The Issues Kanban board is the best tool for quickly adding issues and child tasks. To open, choose **Boards>Boards**.

The screenshot shows the Azure DevOps interface. At the top left is the 'Azure DevOps' logo and the project name 'fabrikam / Fabrikam'. A red circle labeled '1' highlights the project dropdown. On the left sidebar, under the 'Boards' section, 'Boards' is selected and highlighted with a red box, while 'Issues' is also visible. A red circle labeled '2' is on the sidebar. In the center, the 'Fabrikam Team' board is selected, shown by a red box around its dropdown menu. A red circle labeled '3' is on the board header. The board has three columns: 'To Do', 'Doing', and 'Done'. The 'To Do' column contains a 'New item' button and a task 'Secure sign-in' (Priority 1). The 'Doing' column contains a task 'Hello World web site' (Priority 2) and a progress bar showing 0/4 completed. The 'Done' column contains tasks 'Cancel form' (Priority 3) and 'Change background'.

The Epics Kanban board is the best tool for quickly adding epics and issues that are children of those epics. To open the Epics board from the Issues board, choose **Epics** from the board selector.

This screenshot shows the same 'Fabrikam Team' board as above, but with a dropdown menu open over the 'Issues' button in the top right. The dropdown menu has two options: 'Epics' (highlighted with a red box and a red arrow pointing to it) and 'Issues'.

Add issues or user stories

- Basic process
- Agile process

1. From the Issues board, choose **New item** and start adding those issues you want to track.

This screenshot shows the 'Fabrikam Team' board with the 'New item' button in the 'To Do' column highlighted with a red box. The board structure remains the same with 'To Do', 'Doing', and 'Done' columns.

2. Enter return and the system assigns a work item ID to the issue.

3. To track the work you want to manage, add as many issues that you need.

Add details to an issue or user story

Choose the issue or user story title to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop a file to share the file with others.

- [Basic process](#)
- [Agile process](#)

For example, here we assign the issue to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa.

NOTE

You can only assign work to a user who has been added to the project.

Choose **Save & Close** when done.

Field description

FIELD	DEFINITION
Title	Enter a description of 255 characters or less. You can always modify the title later.

Assigned To	Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu will list only team members or contributors to the project.
State	When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current state.
Reason	Use the default first. Update it when you change state as need. Each State is associated with a default reason.
Area	Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see Define area paths and assign to a team .
Iteration	Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see Define iteration paths (aka sprints) and configure team iterations .
Description	Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.
Priority	A subjective rating of the issue or task it relates to the business. You can specify the following values: 1: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible. 2: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately. 3: Resolution of the work item is optional based on resources, time, and risk. 4: Resolution of the work item is not required.
Effort	Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers. The estimates you set for Effort are used to calculate velocity and to forecast sprints .

Update status

- [Basic process](#)
- [Agile process](#)

As work starts, drag the issue from the **To Do** column to the **Doing** column. Once completed, move to the **Done** column.

To Do

- + New item
- Change new item
Priority 2

Doing 4/5

- Secure sign-in
Raisa Pokrovskaya
Priority 1
- Hello World web site
Jamal Hartnett
Priority 2

Done

- Cancel form
Christie Church
Priority 3
- Change background
Raisa Pokrovskaya
Priority 2

You can add or rename columns as needed, see [Customize your board](#).

Add tasks

- [Basic process](#)
- [Agile process](#)

Task checklists provide a quick and easy way to track elements of work which are important to support completing an issue.

1. To start adding tasks, choose the ... actions icon for the issue and select the + **Add Task** option.

To Do

- + New item
- Hello World web site
Jamal Hartnett
Priority 2
- Welcome back page
Christie Church
Priority 2

Doing 4/5

- Change background
Raisa Pokrovskaya

... (context menu)

- Open
- Edit title
- Move to iteration >
- + Add Task** (highlighted with a red box)
- Delete

Done

Enter a title for the task and type Enter when done.

Hello World web site
Jamal Hartnett
Priority 2

+ Add Task

<input checked="" type="checkbox"/>	Design welcome screen
-------------------------------------	-----------------------

2. If you have a number of tasks to add, simply keep typing your task titles and type Enter.

Hello World web site

Jamal Hartnett

Priority 2

0/4

Add Task

- Design welcome screen
- change background color
- Standardize form factors
- About screen

3. You can mark a task as done, expand or collapse the task checklist, or reorder and reparent tasks.

MARK A TASK AS DONE	REORDER AND REPARENT TASKS	EXPAND OR COLLAPSE THE CHECKLIST
<p>To mark a task as complete, check the task checkbox. The task State changes to Done.</p> <p>Hello World web site</p> <p>Jamal Hartnett</p> <p>Priority 2</p> <p>0/4</p> <p>Add Task</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Design welcome screen <input checked="" type="checkbox"/> Change background color <input checked="" type="checkbox"/> Standardize form factors <input checked="" type="checkbox"/> About screen 	<p>To reorder a task, drag it within the checklist. To reparent a task, drag it to another issue on the board.</p> <p>Hello World web site</p> <p>Jamal Hartnett</p> <p>Priority 2</p> <p>0/4</p> <p>Add Task</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Design welcome screen <input type="checkbox"/> Standardize form factors <input checked="" type="checkbox"/> Change background color <input checked="" type="checkbox"/> About screen 	<p>To expand or collapse a task checklist, simply choose the task annotation.</p> <p>Hello World web site</p> <p>Jamal Hartnett</p> <p>Priority 2</p> <p>2/4</p>

Add details to a task

If you have details you want to add about a task, choose the title, to open it. Change one or more field values, add a description, or make a note in the **Discussion** section. Choose **Save & Close** when done.

- [Basic process](#)
- [Agile process](#)

Here we assign the task to Jamal.

TASK 9*

9 Design welcome screen

Jamal Hartnett 0 comments Add tag Save & Close Follow ...

State	To Do	Area	Fabrikam	Updated 9 minutes ago
Reason	Added to bac...	Iteration	Fabrikam\Sprint 1	
				Details Related Work items (1)

Description

Click to add Description

Discussion

Add a comment. Use # to link a work item, ! to link a pull request, or @ to mention a person.

Planning

Priority
2

Activity

Remaining Work

Development

+ Add link
Development hasn't started on this item.

Related Work

+ Add link Parent

7 Hello World web site Updated 29 minutes ago, To Do

Field descriptions

In addition to the fields you can define for an issue, you can specify the following fields for a task to support capacity and time tracking.

FIELD	DEFINITION
Activity	<p>The type of activity that is required to perform a task. To learn more about how this field is used, see Capacity planning. Allowed values are:</p> <ul style="list-style-type: none"> Deployment Design Development Documentation Requirements Testing
Remaining Work	<p>The amount of work that remains to finish a task. You can specify work in hours or in days. There are no inherent time units associated with this field even though the taskboard always shows "h" for hours in relationship to Remaining Work.</p> <p>Remaining Work is often used to calculate burn down for a sprint.</p>

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.

The rich text editor tool bar displays below the text entry area when you click your cursor within each text box that can be formatted.

Mention someone, a group, work item, or pull request (, , or)

Choose one of these icons —, , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or, you can simply type @, #, or ! to open the same menu.

NOTE

This latest version of the rich text editor requires Azure DevOps Server 2019 Update 1 or later version.

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing @ and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the actions icon and then choose **Delete**.

NOTE

The edit/delete feature requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention control** to notify another team member about the discussion. Simply type @ and their name. To reference a work item, use the **#ID control**. Type # and a list of work items that you've recently referenced will appear from which you can select.

To reference a work item, use the **#ID control**. Type # and a list of work items that you've recently referenced will appear from which you can select.

IMPORTANT

For on-premises Azure DevOps Server or TFS, [you must configure an SMTP server](#) in order for team members to receive notifications.

Note that you can't edit or delete comments once they've been entered.

Add a reaction to a comment

You can add one or more reactions to any comment. Choose a smiley icon at the upper-right corner of any comment or choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, click the reaction on the bottom of your comment. The following shows an example of the experience of

adding a reaction, as well as the display of reactions on a comment.



Try this next

[Customize your board](#)

Related articles

- [Index to field descriptions](#)
- [Add tags to issues or tasks](#)
- [Use @mentions in work items](#)
- [Use #ID to link to work items](#)

Quickstart: Add, run, and update inline tests

8/1/2019 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

In this quickstart, you learn how to add, run, update, and expand and collapse inline tests in Azure DevOps.

A quick way to start manual testing is to add the test to the user story or bug that you want to test. From the Kanban board, you can quickly define inline tests, or a set of manual tests for a backlog item. You also can run these tests and update their status. If you're new to working with the Kanban board, see the [Kanban quickstart](#).

Tests you create from the Kanban board are automatically linked to the user story or backlog item.

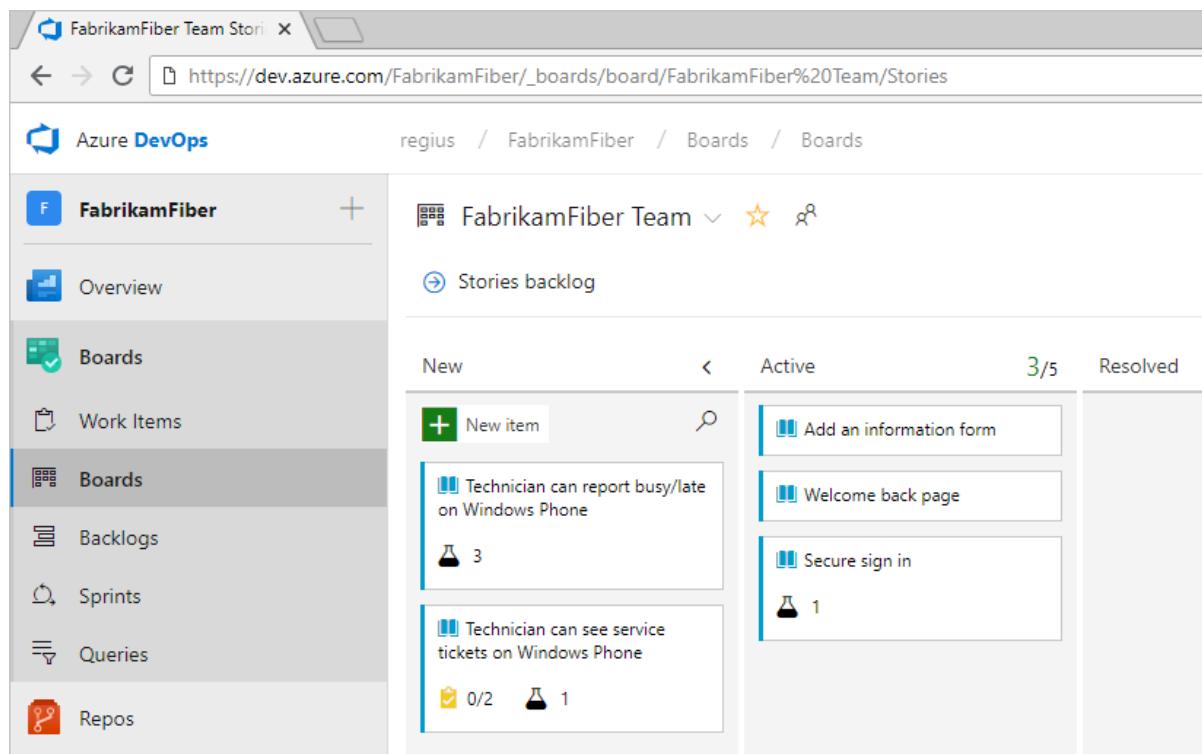
Open your Kanban board

1. From your web browser, open the project for your organization and select **Azure Boards**. If you don't have a project, [create one now](#). If you haven't been added as a team member, [get invited now](#).

The URL follows this pattern: `https://dev.azure.com/fabrikamfiber/_boards/board`

If you don't see the team or project you want, select the  to [browse all projects and teams](#).

2. Select **Boards** to open the Kanban board.



New	Active	Resolved
New item   Technician can report busy/late on Windows Phone  3	Add an information form Welcome back page Secure sign in  1	

1. From your web browser, open the project for your organization and select **Azure Boards**. If you don't have a project, [create one now](#). If you haven't been added as a team member, [get invited now](#).

The URL follows this pattern: `https://dev.azure.com/fabrikamfiber/_backlogs/board`

If you don't see the team or project you want, select the  Azure DevOps icon to [browse all projects and teams](#).

2. Select **Board** to open the Kanban board.

The screenshot shows the 'Backlogs - Azure DevOps' page for the 'FabrikamFiber' project. The left sidebar has 'Backlogs' selected. The main area displays a 'Stories' backlog board. The top navigation bar includes 'Work' and '...'. A search bar at the top right says 'Search work items in this project'. The backlog board has tabs for 'Backlog' and 'Board'. It shows two columns: 'New' (containing one item) and 'Active' (containing three items). The 'Active' column has sub-tiles for 'Add an information form', 'Welcome back page', and 'Secure sign in'. The bottom of the board shows iteration counts: Iteration 1 (3), Iteration 2 (0/2), and Iteration 3 (1).

Add tests

1. To add tests, open the menu for a work item.

The screenshot shows the 'FabrikamFiber Team Stories' page. The left sidebar has 'Boards' selected. The main area displays a 'Stories backlog' board. The top navigation bar includes 'Azure DevOps' and 'Fabrikam / FabrikamFiber / Boards / Boards'. The backlog board shows two items: 'Technician can report busy/late on Windows Phone' and 'Technician can see service tickets on Windows Phone'. A context menu is open over the first item, with the 'Add Test' option highlighted by a red box.

Adding inline tests is the same as adding test cases to a test suite. A default test plan and test suite are automatically created under which the manual test cases are grouped.

For example, a test suite is created for the following user story, and inline tests are added to that suite. User story 314 is highlighted. It has two manual tests defined with the IDs 337 and 341.

The screenshot shows the 'Test Plans' page in the 'FabrikamFiber' project. A test suite named '314 : Technician can see service tickets on Windows Phone' is selected. The suite details are shown on the right, including the title 'Test suite: 314 : Technician can see service tickets on Windows Phone'. Below this is a 'Tests' tab showing a table of two manual tests:

Outcome	Order	ID ↑	Title
Active	1	337	Change colors on initial view
Active	2	341	Change initial page size

2. If you have a number of tests to add, enter each title and select **Enter**.

The screenshot shows the 'Add Test' dialog for the test case '314 : Technician can see service tickets on Windows Phone'. The dialog lists two existing tests: 'Change colors on initial view' and 'Change initial page size'. A new test entry is being typed into the text input field at the bottom.

To add details to the test case, open it. You can select the title, double-select the inline item, or open the context menu and choose **Open**.

The screenshot shows the details of a test case named '337 Change colors on initial view'. The top navigation bar includes 'TEST CASE 337', 'Jamal Hartnett' (owner), '0 comments', 'Add tag', 'Save & Close', 'Follow', and a '...' button. Below this, the 'State' is 'Design' and 'Reason' is 'New'. The 'Area' is 'FabrikamFiber' and the 'Iteration' is 'FabrikamFiber\Release 1\Sprint 3'. A tabs bar at the bottom includes 'Steps' (selected), 'Summary', 'Associated Automation', and buttons for 'Edit' (with a clock icon), 'History' (with a circular arrow icon), and 'Comments' (with a speech bubble icon). The main area has a 'Steps' section with a placeholder 'Click or type here to add a step' and a toolbar with icons for file operations. To the right, there's a 'Development' section with a note 'Development hasn't started on this item.' and a 'Related Work' section listing a 'Technician can see service' task. The 'Details' section shows 'Priority' as 2 and 'Automation status' as 'Not Automated'. At the bottom, there's a 'Parameter values' section.

To learn more about how to define tests, see [Create manual tests](#).

Before you run the test, you must add details.

1. To add tests, open the menu for the work item.

The screenshot shows a 'Test Plan' interface with three columns: 'Analyze', 'Code', and 'Test'. In the 'Analyze' column, a user story '168 Hello World Web Site' is selected. A context menu is open over it, with the 'Add Test' option highlighted with a blue selection bar. Other options in the menu include 'Open', 'Edit title', 'Add Task', 'Delete', 'Do exploratory testing', and 'New branch'. The 'Code' and 'Test' columns are also visible.

Adding inline tests is the same as adding test cases to a test suite. A default test plan and test suite are automatically created under which the manual test cases are grouped.

For example, a test suite is created for each user story, and all inline tests are added to that suite. The following user story 152 is highlighted. It has three manual tests defined with the IDs 153, 155, and 161.

The screenshot shows the 'Test plan' section of a software application. At the top, there are tabs for HOME, CODE, WORK, BUILD, and TEST. Below the tabs, the 'Test plan' tab is selected, along with 'Parameters', 'Runs', and 'Machines*'. A search bar contains the text 'FabrikamFiber: FabrikamFiber Team_Stories_FabrikamFiber (Id: 157)'. On the left, a tree view shows a project structure: 'FabrikamFiber Team_Stories_FabrikamFiber' expanded, with three items under it: '151 : Customer log in (2)', '152: Customer welcome page (3)' (which is selected and highlighted in blue), and '169 : Slow response on form (1)'. On the right, a table titled 'Tests' lists three entries:

Outcome	ID	Title
Active	153	Change colors on initial view
Active	155	Change initial page size
Active	161	Log in with email

To learn more about test plans and test suites, see [Plan your tests](#).

2. If you have a number of tests to add, enter each title and select **Enter**.

This screenshot shows a detailed view of a test case. The title is '152 Customer welcome page'. It includes a user profile picture for 'Raisa Pokrovskaya'. Below the title, there are two progress bars: one yellow labeled '0/3' and one orange labeled '0/3'. A 'Add Test' button is present. A list of three test cases is shown, each with a blue circular icon:

- Change colors on initial view
- Change initial page size
- Log in with email

To add details to the test case, open it. You can select the title, double-select the inline item, or open the context menu and choose **Open**.

[TEST CASE 153](#)

153 Change colors on initial view

Design Raisa Pokrovskaya 0

Area Iteration
Fabrikam Fiber Fabrikam Fiber\Iteration 1

Add Tag Steps Summary Associated Automation (1)

Steps

Action Click or type here to add a step

Development

Development hasn't started on this [Create a new branch](#)

Status

Priority 2

Automation status Not Automated

Parameter values [Add a shared parameter set](#) | [Convert to shared parameters](#)

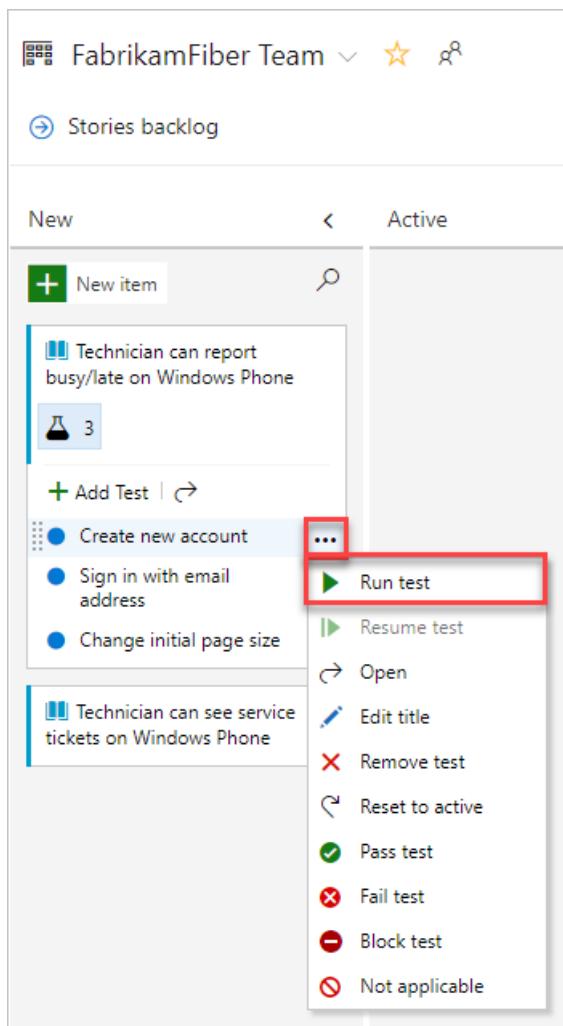
The screenshot shows a test case named '153 Change colors on initial view' under the 'Design' category. It's assigned to 'Raisa Pokrovskaya' and has 0 comments. The 'Area' is 'Fabrikam Fiber' and the 'Iteration' is 'Fabrikam Fiber\Iteration 1'. A 'Steps' tab is selected, showing a placeholder 'Click or type here to add a step'. To the right, there's a 'Development' section indicating no development has started yet, with a link to 'Create a new branch'. Below it is a 'Status' section with priority set to 2 and automation status as 'Not Automated'. At the bottom, there's a 'Parameter values' section with links to add a shared parameter set or convert to shared parameters.

To learn more about how to define tests, see [Create manual tests](#).

Before you run the test, you must add details.

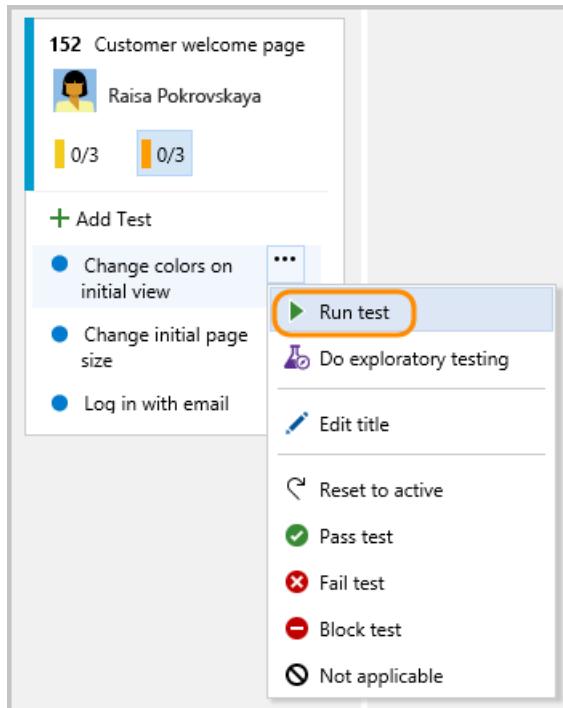
Run a test

Run the test by selecting **Run test** from the actions menu for the inline test.



Microsoft Test Runner starts in a new browser instance. For information on how to run a test, see [Run manual tests](#).

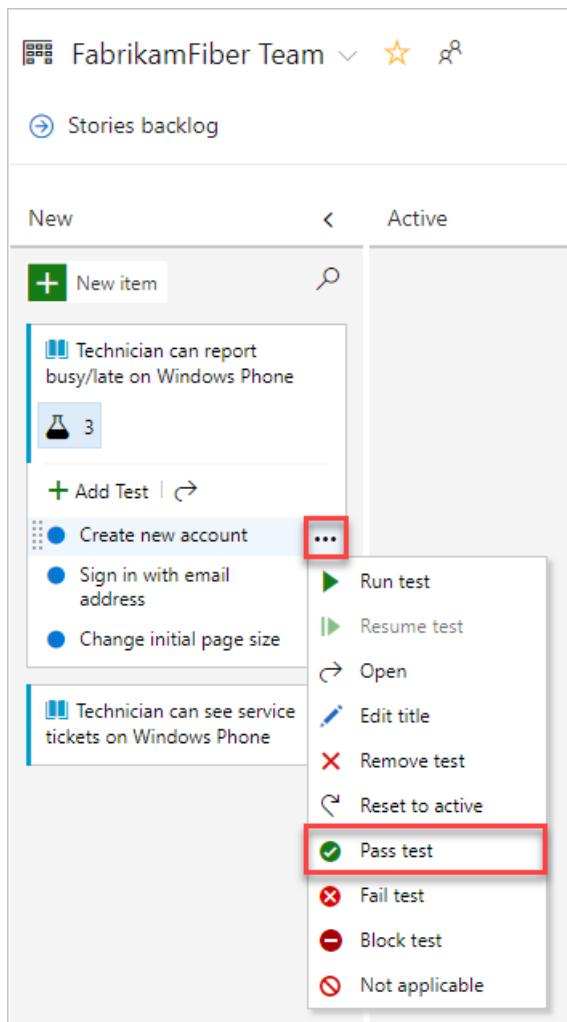
Run the test by selecting **▶ Run test** from the **...** actions menu for the inline test.



Microsoft Test Runner starts in a new browser instance. For information on how to run a test, see [Run manual tests](#).

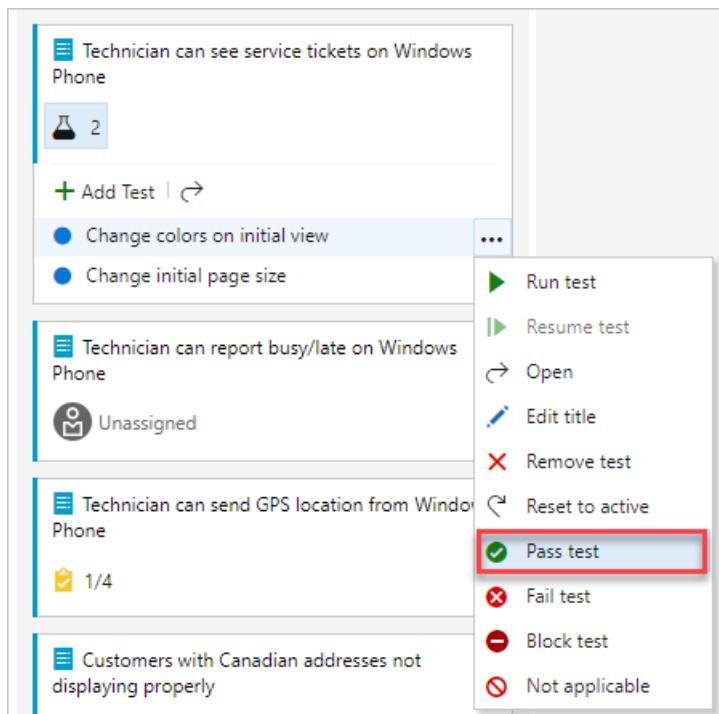
Update the status of a test

You can update the status of the test from the *** actions menu.



When you update the status of tests, you can [track test results](#).

You can update the status of the test from the *** actions menu.



When you update the status of tests, you can [track test results](#).

Expand or collapse inline tests

When you first open the Kanban board, you'll see an unexpanded view of checklists and tests.

The screenshot shows the Azure DevOps interface for the 'FabrikamFiber' project. On the left, the sidebar is open with 'Boards' selected. The main area displays the 'Stories backlog' for the 'FabrikamFiber Team'. A specific story card is shown in detail:

New	Active	Resolved
New item	Add an information form	
Technician can report busy/late on Windows Phone 3	Welcome back page	
Technician can see service tickets on Windows Phone 0/2 1	Secure sign in 1	

The 'Technician can see service tickets on Windows Phone' card has its checklist items collapsed, indicated by the '0/2' status. A red arrow points from the collapsed state to the expanded state below, where the checklist items are visible.

Select the inline test summary to expand a collapsed set of tests. Select the same summary to collapse an expanded list.

This screenshot illustrates the interaction with inline test summaries. On the left, a story card for 'Technician can see service tickets on Windows Phone' shows a checklist with one item, 'Jamal Hartnett', and a status of '2'. A red arrow points from this collapsed state to the expanded state on the right, where the same checklist is shown with the same details and a status of '2'.

When you first open the Kanban board, you'll see an unexpanded view of checklists.

Backlog **Board**

	Analyze	Code	Test
New	<p>168 Hello World Web Site  Jamal Hartnett 1/4</p> <p>152 Customer welcome page  Raisa Pokrovskaya 0/3 0/3</p>	<p>151 Customer log in  Christie Church 0/3</p>	<p>169 Slow response on form  Raisa Pokrovskaya 0/1</p>

Select the inline test summary to expand a collapsed set of tests. Select the same summary to collapse an expanded list.

Code **1/5**

151 Customer log in
 Christie Church
0/3



Code **1/5**

151 Customer log in ...
 Christie Church
0/3

+ Add Test

- Log in with email
- Create new account
- Change initial page size

Next steps

[Learn more about test case management Kanban quickstart](#)

To begin web-based exploratory testing for a user story, you must install the Exploratory testing. For more information, see how to [Exploratory test your web app directly in your browser](#).

View permissions for yourself or others

9/3/2019 • 3 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this quickstart, you learn how to view your permissions or the permissions that are set for others in Azure DevOps. If you don't have a permission to access a feature or function, you can request it from the right resource.

Permissions are set at the collection, project, and object level as described in [About permissions and groups](#). So to view the permissions you have, you need to open the permissions at the object, project, or collection level.

Prerequisites

- You must have a project to connect to. If you don't have a project yet, [create one](#).
- You must be a member of the Project Valid Users Group or Project Collection Valid Users Group to view permissions.

NOTE

This article shows how to view permissions assigned to a user at the project-level or collection-level. However, the steps are similar when you work from the Security dialog of an object.

View project-level permissions

NOTE

To enable the new user interface for the Project Permissions Settings Page, see [Enable preview features](#).

- [Preview page](#)
- [Current page](#)

1. Choose **Project Settings** and then **Permissions**.

The screenshot shows the 'Groups' section of the 'Permissions' tab in the Project Settings. The 'Users' tab is active, indicated by a red border. A search bar at the top right contains the prefix 'Jamal'. Below it is a table listing project groups:

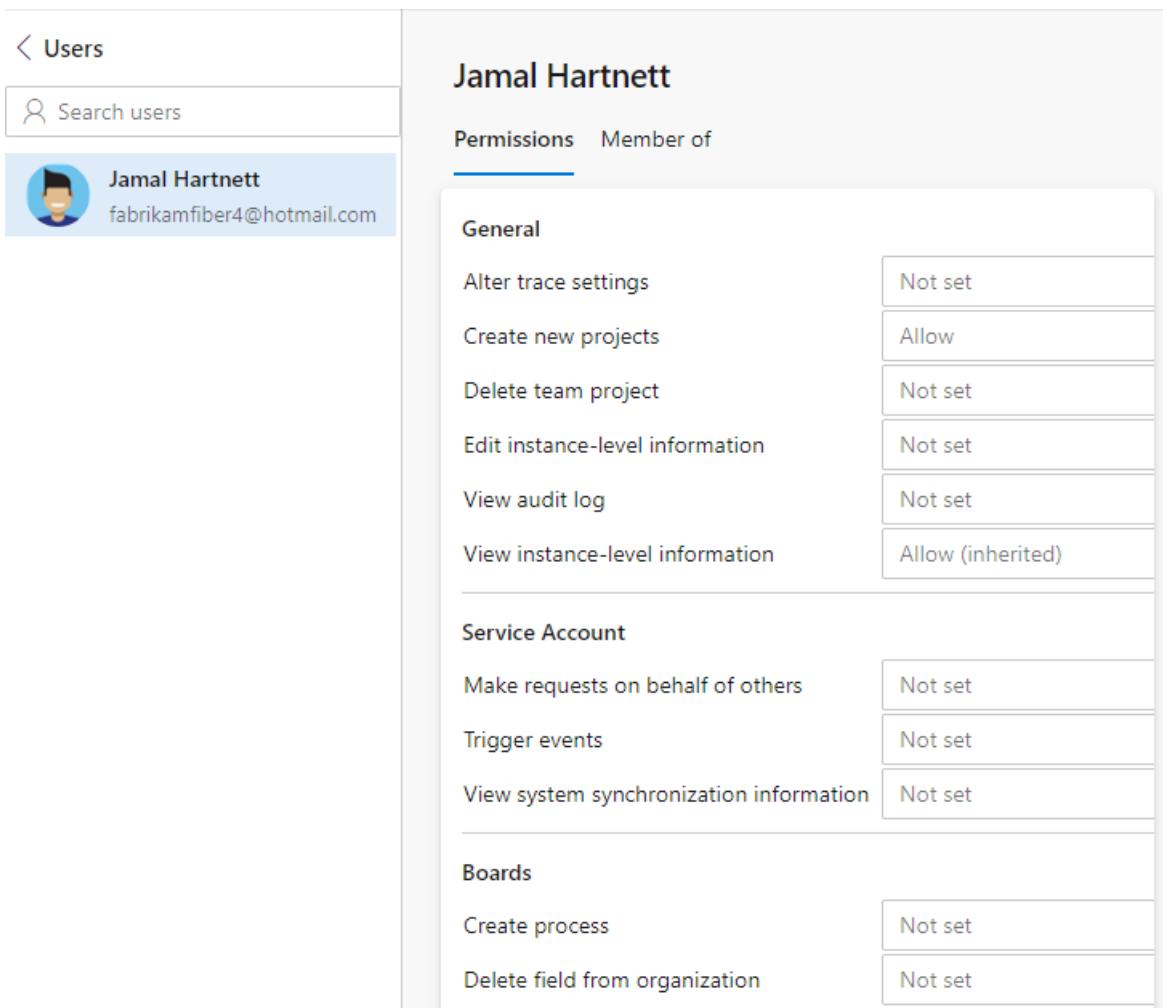
Name	Description	Type	Members
[Fabrikam Fiber]\Build Administrators	Members of this group ... Group	Group	0
[Fabrikam Fiber]\Contributors	Members of this group ... Group	Group	3
[Fabrikam Fiber]\Project Administrators	Members of this group ... Group	Group	1
[Fabrikam Fiber]\Project Valid Users	Members of this group ... Group	Group	6
[Fabrikam Fiber]\Readers	Members of this group ... Group	Group	0
[Fabrikam Fiber]\Fabrikam Fiber Team	The default project team. Team	Team	2
[Fabrikam Fiber]\Web		Team	1

2. Choose **Users**. To filter the list, enter a name into the *Search groups or users* box.

The screenshot shows the 'Permissions' - 'Users' page. The 'Users' tab is active, indicated by a red border. A search bar at the top right contains the prefix 'Jamal'. A list of users is shown below, with 'Jamal Hartnett' highlighted in a dropdown menu:

Name
Jamal Hartnett fabrikamfiber4@hotmail.com
AB Azure Boards
MS MyPublicProject Build Service (fabrikam)
DS Demo 11 Build Service (fabrikam)
Christie Church fabrikamfiber1@hotmail.com
CR Chuck Reinhart fabrikamfiber3@hotmail.com
Jamal Hartnett fabrikamfiber4@hotmail.com

3. Choose the name you want. The project-level permissions for that user displays. These permissions are based on the groups the user belongs to or the permissions set specifically for the user's account.



The screenshot shows the 'Users' section of the Microsoft Teams admin center. On the left, there's a search bar and a list of users. The user 'Jamal Hartnett' is selected, showing their details on the right. The 'Permissions' tab is active, displaying a table of permissions and their status. The 'Member of' tab is also visible at the top.

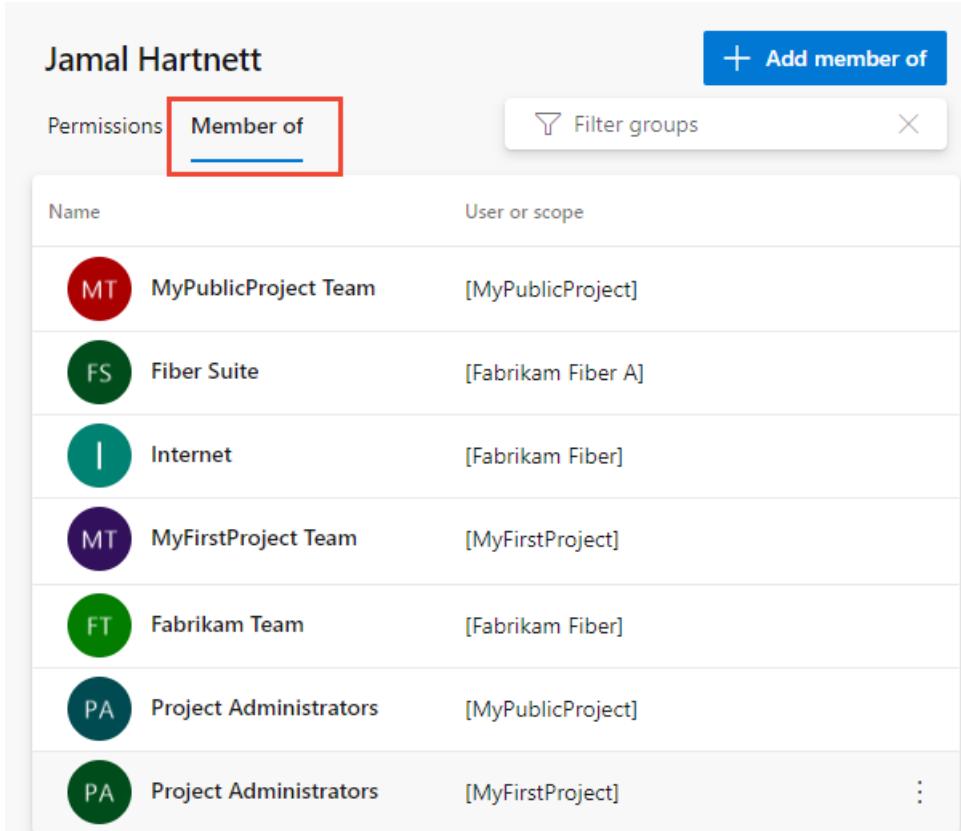
Permission	Status
Alter trace settings	Not set
Create new projects	Allow
Delete team project	Not set
Edit instance-level information	Not set
View audit log	Not set
View instance-level information	Allow (inherited)

Permission	Status
Make requests on behalf of others	Not set
Trigger events	Not set
View system synchronization information	Not set

Permission	Status
Create process	Not set
Delete field from organization	Not set

4. Choose **Member of** to see which security groups and teams that the user belongs to.

Here we see that *Jamal Hartnett* belongs to several teams and the Project Collection Administrators group for several projects.



The screenshot shows the 'Member of' tab for the user 'Jamal Hartnett'. It lists various teams and security groups he is a member of, each with a small icon and the name of the group and its scope.

Name	User or scope
MyPublicProject Team	[MyPublicProject]
Fiber Suite	[Fabrikam Fiber A]
Internet	[Fabrikam Fiber]
MyFirstProject Team	[MyFirstProject]
Fabrikam Team	[Fabrikam Fiber]
Project Administrators	[MyPublicProject]
Project Administrators	[MyFirstProject]

1. Open **Project Settings**. Choose the gear settings icon, and choose **Security**.

The screenshot shows the 'Fabrikam Fiber' project settings. The top navigation bar includes 'Fabrikam Fiber', 'Dashboards', 'Code', 'Work', '...', and a gear icon for 'Settings'. Below the navigation is a secondary menu with tabs: 'Overview', 'Work', 'Security' (which is highlighted with a red box), 'Version Control', 'Policies', 'Agent queues', 'Notifications', and 'Service Hooks'.

2. Begin entering the name into the *Filter users and groups* box. The system automatically shows the names that begin with the characters you enter.

The screenshot shows the 'Create group' screen. On the left, a search results panel displays 'Showing 1 result' for 'Jamal'. It shows a card for 'Jamal Hartnett' with a blue profile picture and the email 'fabrikamfiber4@hotmail.com'. On the right, the main pane shows 'Fabrikam Fiber > Customer Service' with permission settings for 'Jamal Hartnett'. The permissions listed are: Create tag definition (Allow (inherited)), Create test runs (Allow (inherited)), Delete and restore work items (Not set), Delete team project (Not set), Delete test runs (Allow (inherited)), Edit project-level information (Not set), and Manage project properties (Not set). Navigation tabs at the top of the right pane include 'Permissions', 'Members', and 'Member of'.

3. Choose the name you want. The project-level permissions you have set are based on the groups you belong to or the permissions set for your account.

The screenshot shows the 'Create group' screen. On the left, a search results panel displays 'Showing 1 result' for 'Jamal'. It shows a card for 'Jamal Hartnett' with a blue profile picture. On the right, the main pane shows 'fabrikam > Jamal Hartnett' with permission settings for 'Jamal Hartnett'. The permissions listed are: Create tag definition (Allow (inherited)), Create test runs (Allow (inherited)), Delete and restore work items (Not set), Delete team project (Not set), Delete test runs (Allow (inherited)), Edit project-level information (Not set), Manage project properties (Not set), Manage test configurations (Allow (inherited)), Manage test environments (Allow (inherited)), Move work items out of this project (Not set), Permanently delete work items (Not set), Rename team project (Not set), View project-level information (Allow (inherited)), and View test runs (Allow (inherited)). Navigation tabs at the top of the right pane include 'Permissions' and 'Member of'.

For a description of each permission, see [Permissions and groups reference](#).

4. Choose **Member of** to see which security groups the user belongs to.

Here we see that *Jamal Hartnett* belongs to several teams and the Project Collection Administrators group.

The screenshot shows the 'Member of' tab selected for the user 'Jamal'. It lists the groups 'Customer Service', 'Fabrikam Fiber Team', 'Web', and 'Project Collection Administrators' under 'fabrikam > Jamal Hartnett'. A red box highlights the 'Member of' tab.

Display Name	Username Or Scope	
Customer Service	[Fabrikam Fiber]	Remove
Fabrikam Fiber Team	[Fabrikam Fiber]	
Web	[Fabrikam Fiber]	
Project Collection Administrators	[fabrikam]	

For a description of each group, see [Permissions and groups reference](#).

View organization or collection-level permissions

Open admin settings for the organization or a project collection.

NOTE

To enable the new user interface for the Organizations Permissions Settings Page v2, see [Enable preview features](#). The preview page provides a group settings page that the current page does not.

1. Choose the to open **Projects**. Then choose **Organization settings**.



2. Choose **Permissions**, the **Project Collection Administrators** group, and then **Members**.

The screenshot shows the 'Members' tab selected for the 'Project Collection Administrators' group. It lists five members: Christie Church, Project Collection Service Accounts, Jamal Hartnett, Raisa Pokrovskaya, and Helena Petersen. A red box highlights the 'Members' tab.

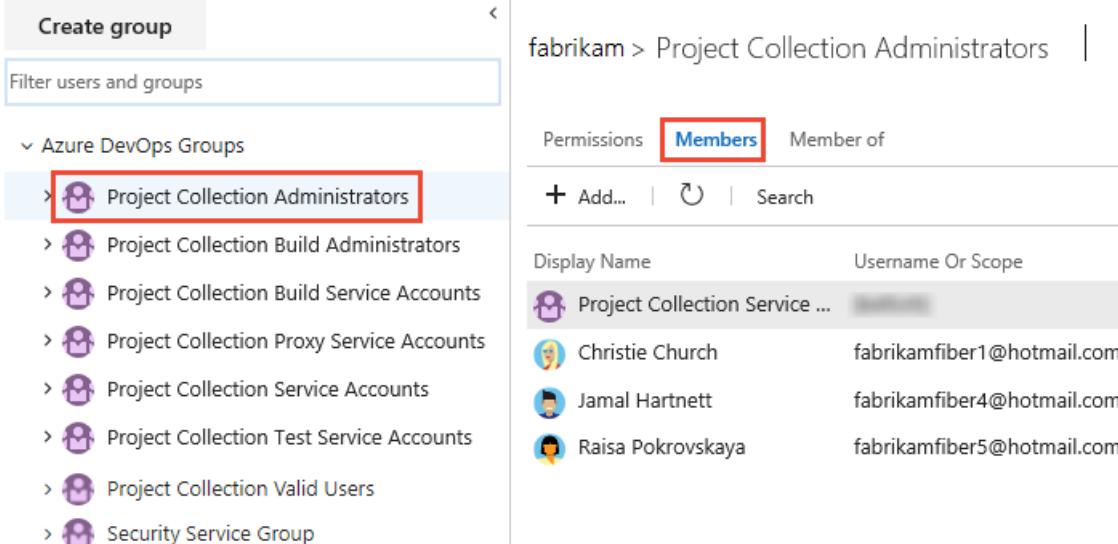
Name	Type	Username or scope
Christie Church fabrikamfiber1@hotmail.com	user	fabrikamfiber1@hotmail.com
PA Project Collection Service Accounts	group	[mseng]
Jamal Hartnett fabrikamfiber4@hotmail.com	user	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya fabrikamfiber5@hotmail.com	user	fabrikamfiber5@hotmail.com
Helena Petersen fabrikamfiber8@hotmail.com	user	fabrikamfiber8@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

1. Choose the to open **Projects**. Then choose **Admin settings**.



2. Choose **Security**, the **Project Collection Administrators** group, and then **Members**.



The screenshot shows the 'Create group' interface. On the left, under 'Azure DevOps Groups', the 'Project Collection Administrators' group is highlighted with a red box. On the right, the 'Members' tab is selected, showing a list of users and their email addresses:

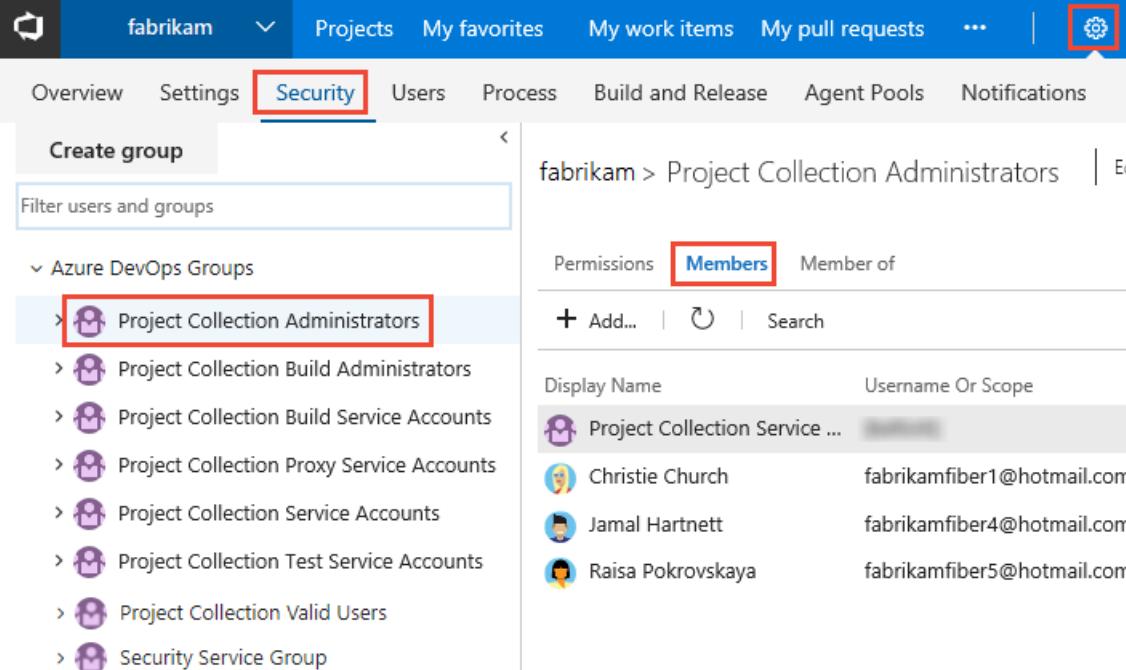
Display Name	Username Or Scope
Project Collection Service ...	[REDACTED]
Christie Church	fabrikamfiber1@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

1. Choose the settings icon and select **Organization settings** or **Collection settings**.



2. Choose **Security**, **Project Collection Administrators** group, and then **Members**.



The screenshot shows the main Azure DevOps navigation bar with the 'Security' tab highlighted with a red box. Below it, the 'Project Collection Administrators' group is highlighted with a red box. On the right, the 'Members' tab is selected, showing the same user list as the previous screenshot:

Display Name	Username Or Scope
Project Collection Service ...	[REDACTED]
Christie Church	fabrikamfiber1@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com

3. Follow steps 2 through 4 in the procedure outlined previously for view project-level permissions.

View object-level permissions

You can define the security or permissions for a number of objects. You access them from the context menu of the object.

From the web portal, open the Security dialog for the object whose permissions you want to set. For specific instructions, see the following articles:

AREA	TASK

Wiki & Dashboard permissions	<ul style="list-style-type: none">• README & Wiki• Dashboards
Azure Repos, Azure Pipelines/DevOps (code, build, test, release) permissions	<ul style="list-style-type: none">• Git branch• Git repository• TFVC• Builds• Release pipeline security• Approvals and approvers
Azure Boards/Work tracking permissions	<ul style="list-style-type: none">• Area and iteration paths• Work item query and folder• Plan permissions

Next steps

Look up the organization owner or a Project Administrator

Quickstart: Sign up, sign in to Azure DevOps

8/1/2019 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services

In this quickstart, you learn how to sign up for Azure DevOps for free, sign in with a Microsoft or GitHub account, create an organization, create a project, and invite your teammates.

Sign up for Azure DevOps to upload and share code in free, unlimited private Git repositories.

Then, connect to your favorite development tool like Eclipse, Xcode, Visual Studio, IntelliJ, or Android Studio to work on apps anytime, anywhere.

Sign up with a personal Microsoft account

1. Select the sign-up link for [Azure DevOps](#).
2. Enter your email address, phone number, or Skype ID for your Microsoft account. If you're a Visual Studio subscriber and you get Azure DevOps as a benefit, use the Microsoft account associated with your subscription. Select **Next**.

If you don't have a Microsoft account, choose **Create one**. To learn more, see [create a Microsoft account](#).

3. Enter your password and select **Sign in**.

4. To get started with Azure DevOps, select **Continue**.

An organization is created based on the account you used to sign in. Sign in to your organization at any time, (<https://dev.azure.com/{yourorganization}>).

You can rename and delete your organization, or change the organization location. To learn more, see the following articles:

- [Rename an organization](#)
- [Change the location of your organization](#)

If you signed in with an existing Microsoft account, your next step is to [Create a project](#). If you signed in with a newly created Microsoft account, then your project is automatically created and named after your account name. To learn more about managing projects, see [Manage projects](#).

Sign up with a GitHub account

IMPORTANT

If your GitHub email address is associated with an Azure AD-backed organization in Azure DevOps, you can't sign in with your GitHub account, rather you must sign in with your Azure AD account.

1. Select the sign-up link for [Azure DevOps](#), **Start free with GitHub**. If you're already part of an Azure

DevOps organization, select **Sign in to Azure DevOps**.

2. Select **Sign in with GitHub**.

If you have an account in session already, select **Use another account**. You're taken to GitHub sign-in where you can enter your GitHub user name or email address.

3. Enter your GitHub account credentials, and then select **Sign in**.

4. Select **Authorize Microsoft corporation**.

5. To get started with Azure DevOps, select **Continue**.

An organization is created based on the account you used to sign in. Sign in to your organization at any time, (

<https://dev.azure.com/{yourorganization}>).

You can rename and delete your organization, or change the organization location. To learn more, see [Manage organizations](#).

Enable GitHub invitations

Creating a new Azure DevOps organization with your GitHub username turns on the Invite GitHub users policy by default. For existing organizations, your administrator can turn on this capability via **Organization settings > Policies** tab.

Once the setting is changed, sign out of Azure DevOps, and then from a fresh browser session, sign back in to the organization `dev.azure.com/{organizationName}` or `organizationName.visualstudio.com` with your GitHub credentials. You're now recognized as a GitHub user and the GitHub invitation experience is available to you.

The screenshot shows the Azure DevOps Organization Settings Policies page. On the left, there's a sidebar with 'General' and 'Security' sections. Under 'Security', the 'Policies' tab is selected and highlighted with a red box. In the main area, under 'Application connection policies', three settings are listed: 'Alternate authentication credentials' (On), 'Third-party application access via OAuth' (On), and 'SSH authentication' (On). Under 'Security policies', 'Allow public projects' is set to On. Under 'User policies', 'Invite GitHub users' is set to Off and is also highlighted with a red box.

For more information about GitHub authentication, see [FAQs](#).

Create a project

If you signed up for Azure DevOps with a newly created Microsoft account (MSA), your project is automatically created and named based on your sign-in.

If you signed up for Azure DevOps with an existing MSA or GitHub identity, you're automatically prompted to create a project. You can create either a public or private project. To learn more about public projects, see [What is a public project?](#).

1. Enter information into the form provided, which includes a project name, description, visibility selection, initial source control type, and work item process.

See [choosing the right version control for your project](#) and [choose a process](#) for guidance.

2. When your project is complete, the welcome page appears.

Invite team members

Give team members access to your organization by adding their email addresses or GitHub usernames to your organization. For GitHub user invitations, ensure you've enabled the policy, [Invite GitHub users](#) in **Organization settings > Policies** tab.

1. Sign in to your organization (<https://dev.azure.com/{yourorganization}>).
2. Select **Organization settings**.

3. Select **Users > Add new users**.

4. Enter the following information:

- **Users:** Enter the email addresses (Microsoft accounts) or [GitHub usernames](#) for the users. You can add several email addresses by separating them with a semicolon (;). An email address appears in red when it's accepted.
- **Access level:** Leave the access level as **Basic** for users who will contribute to the code base. To learn more, see [About access levels](#).
- **Add to project:** Select the project you want to add them to.
- **DevOps Groups:** Leave as **Project Contributors**, the default security group for users who will contribute to your project. To learn more, see [Default permissions and access assignments](#).

NOTE

Add email addresses for [personal Microsoft accounts](#) and IDs for GitHub accounts unless you plan to use [Azure Active Directory \(Azure AD\)](#) to authenticate users and control organization access. If a user doesn't have a Microsoft or GitHub account, ask the user to [sign up](#) for a Microsoft account or a GitHub account.

5. When you're done, select **Add** to complete your invitation.

For more information about managing users and organization access, see [Add organization users for Azure DevOps](#).

Next steps

[Add code to your Git repository](#)

[Plan and track work](#)

Quickstart: Create an organization or project collection

8/1/2019 • 2 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this quickstart, you learn how to create an organization. An organization is used to connect groups of related projects, helping to scale up an enterprise. You can use a personal Microsoft account, GitHub account, or a work or school account. Use your work or school account to *automatically connect* your organization to your Azure Active Directory (Azure AD).

Prerequisites

1. Read and understand how to [Plan your organizational structure](#).
2. Complete the following steps if you want to use only Microsoft accounts with your organization.

Without Azure AD, you're solely responsible for controlling organization access. And all users must sign in with their Microsoft account. [What are other differences?](#)

- If you don't have a Microsoft account, you can create one when you sign up for Azure DevOps.
 - Use your Microsoft account if you don't need to authenticate users for an organization with [Azure AD](#). All users must sign in to your organization with a Microsoft account.
3. Complete the following steps if you want to authenticate users and control organization access through your Azure AD.
 - You need a work or school account that's managed by your Azure AD. If you use Azure or Office 365, you might have one already. If you don't, learn how to [sign up for Azure as an organization](#).
 - To use existing on-premises identities, see [use Azure AD Connect for integrating on-premises directories with Azure AD](#).
 - All users must be members in that directory to access your organization. To add users from other organizations, use [Azure AD B2B collaboration capabilities](#).

Create an organization

1. Sign in to [Azure DevOps](#).
2. Select **New organization**.

3. Confirm information, and then select **Continue**.

Congratulations, you're now an organization owner!

Sign in to your organization at any time, <https://dev.azure.com/{yourorganization}>.

Create a project collection



A project collection is a container of projects. By grouping projects together, you can manage projects more efficiently and assign the same resources to them.

For more information about how to create a project collection, see [create a project collection](#).

Next steps

[Create a project](#)

Manage your project

9/3/2019 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

With most Azure DevOps services, you can start using the service and configure resources as you go. No up-front work is required. Most settings define defaults.

That said, as an organization Owner or a Project Administrator, there are a few tasks you might want to do once you created your project to ensure a smooth operational experience. If you own a large organization, you'll want to consider additional tasks to structure your projects to support multiple teams or software development apps.

Add users to your project

The first task is to ensure that all members of your organization or group are added to your organization and projects. For small groups, using [Microsoft Accounts](#) to add users to your organization and projects works fine.

Larger enterprises may want to consider using Azure Active Directory to manage permissions and user access. To learn more, see the following articles:

- [Add organization users for Azure DevOps Services](#)
- [Manage user access through Azure Active Directory](#)

The first task is to ensure that all members of your organization or group are added to your organization and project. Larger organizations may want to consider using Azure Active Directory to keep the maintenance of managing permissions and user access. Typically, you should install Azure Active Directory prior to installing TFS.

To learn more, see these articles:

- [Install Azure Active Directory Domain Services \(Level 100\)](#)
- [Step-By-Step: Setting up Azure Active Directory in Windows Server 2016](#)

You can delegate the task to add users to an organization by adding a user with Stakeholder or higher access to the [Project Collection Administrators group](#).

Grant or restrict permissions

Access to features and functions is controlled by access-level assignments, permissions, and security groups. To quickly understand the defaults configured for your project, see [Default permissions and access](#).

If you decide that you want to delegate specific tasks to others, then you'll want to add them to a built-in or custom security group or add them to a specific role. To learn more, see these articles:

- [Grant or restrict access to select features and functions](#)
- [Set permissions at the project level or project collection level](#)

To learn more about permissions and security, review the following articles:

- [About security and identity](#)
- [About permissions and groups](#)
- [About security roles](#)
- [About access levels](#)

Share your project vision and support collaboration

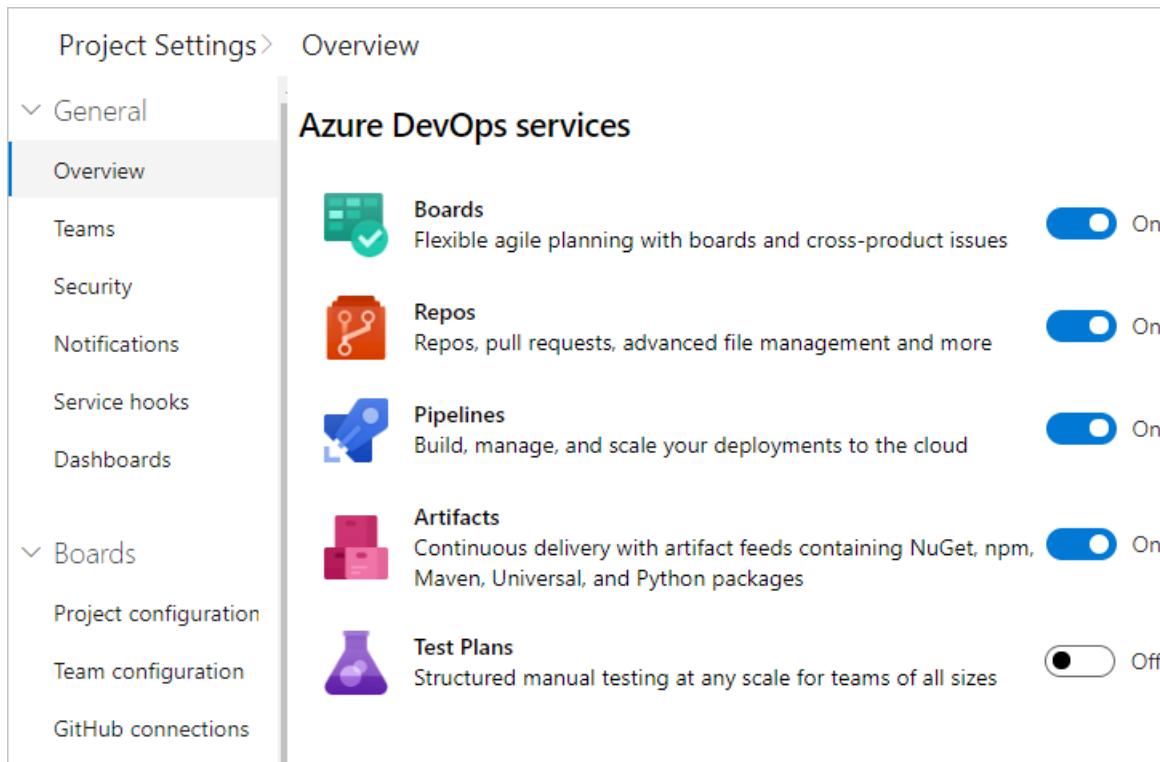
Each project has a summary page where you can share information through **README** files or by pointing to a project Wiki. To orient users who are new to your project and share established processes and procedures, we recommend that you [set up your project summary page](#) or [provision a Wiki](#).

Each project has a summary page where you can share information through **README files**. To orient users who are new to your project and share established processes and procedures, we recommend that you [set up your project summary page](#).

Remove unused services from the user interface

To simplify the web portal user interface, you can disable select services. For example, if you use a project only to log bugs, then you can remove all services except for **Boards**.

This example shows that **Test Plans** has been disabled:



The screenshot shows the 'Project Settings > Overview' page. On the left, a sidebar lists 'General' (selected), 'Teams', 'Security', 'Notifications', 'Service hooks', 'Dashboards', and 'Boards'. Under 'Boards', there are links for 'Project configuration', 'Team configuration', and 'GitHub connections'. The main area is titled 'Azure DevOps services' and contains six service cards with toggle switches:

- Boards**: Flexible agile planning with boards and cross-product issues. Toggled On.
- Repos**: Repos, pull requests, advanced file management and more. Toggled On.
- Pipelines**: Build, manage, and scale your deployments to the cloud. Toggled On.
- Artifacts**: Continuous delivery with artifact feeds containing NuGet, npm, Maven, Universal, and Python packages. Toggled On.
- Test Plans**: Structured manual testing at any scale for teams of all sizes. Toggled Off.

Set code, test, and other policies

There are several policies you can set to support collaboration across your teams, secure your projects, and automatically remove files that are no longer needed. To set policies, review the following articles:

- [Change application access policies for your organization](#)
- [Manage branch policies](#)
- [Add Team Foundation Version Control \(TFVC\) check-in policies](#)
- [Set build and release pipeline retention policies](#)
- [Set test retention policies](#)
- [Manage branch policies](#)
- [Add TFVC check-in policies](#)
- [Set build and release pipeline retention policies](#)
- [Set test retention policies](#)

Define area and iteration paths for work tracking

If you support several products or feature areas, you can assign work items by feature area by setting up [area paths](#). To assign work items to specific time intervals, also known as sprints, you'll want to configure [iteration paths](#). To use the Scrum tools—sprint backlogs, taskboards, and team capacity—you need to configure several sprints. For an overview, see [About areas and iteration paths](#).

Iterations	Areas																																													
<p>Iterations Areas</p> <p>Create and manage the iterations for this project. These are for iteration planning (sprint planning).</p> <p>To access the default team's iteration settings, click here.</p> <p>New New child </p> <table><thead><tr><th>Iterations</th><th>Start Date</th><th>End Date</th></tr></thead><tbody><tr><td>▲ Fabrikam Fiber</td><td></td><td></td></tr><tr><td> ▲ Release 1</td><td></td><td></td></tr><tr><td> Sprint 1</td><td>6/11/2018</td><td>6/29/2018</td></tr><tr><td> Sprint 2</td><td>7/2/2018</td><td>7/20/2018</td></tr><tr><td> Sprint 3</td><td>7/16/2018</td><td>8/3/2018</td></tr><tr><td> Sprint 4</td><td>7/23/2018</td><td>8/10/2018</td></tr><tr><td> Sprint 5</td><td>9/17/2018</td><td>10/5/2018</td></tr><tr><td> Sprint 6</td><td>10/29/2018</td><td>11/16/2018</td></tr><tr><td> Release 2</td><td></td><td></td></tr><tr><td> Release 3</td><td></td><td></td></tr></tbody></table>	Iterations	Start Date	End Date	▲ Fabrikam Fiber			▲ Release 1			Sprint 1	6/11/2018	6/29/2018	Sprint 2	7/2/2018	7/20/2018	Sprint 3	7/16/2018	8/3/2018	Sprint 4	7/23/2018	8/10/2018	Sprint 5	9/17/2018	10/5/2018	Sprint 6	10/29/2018	11/16/2018	Release 2			Release 3			<p>Iterations Areas</p> <p>Create and manage the areas for this project. These are for the team's backlog and what work items the team is tracking.</p> <p>To access the default team's area settings, click here.</p> <p>New New child </p> <table><thead><tr><th>Areas</th><th>Teams</th></tr></thead><tbody><tr><td>▲ Fabrikam Fiber</td><td>Fabrikam Fiber Team</td></tr><tr><td> Customer Service</td><td>Customer Service Team</td></tr><tr><td> Phone</td><td>Fabrikam Fiber Team, Phone</td></tr><tr><td> Voice</td><td>Voice</td></tr><tr><td> Web</td><td>Fabrikam Fiber Team, Web</td></tr></tbody></table>	Areas	Teams	▲ Fabrikam Fiber	Fabrikam Fiber Team	Customer Service	Customer Service Team	Phone	Fabrikam Fiber Team, Phone	Voice	Voice	Web	Fabrikam Fiber Team, Web
Iterations	Start Date	End Date																																												
▲ Fabrikam Fiber																																														
▲ Release 1																																														
Sprint 1	6/11/2018	6/29/2018																																												
Sprint 2	7/2/2018	7/20/2018																																												
Sprint 3	7/16/2018	8/3/2018																																												
Sprint 4	7/23/2018	8/10/2018																																												
Sprint 5	9/17/2018	10/5/2018																																												
Sprint 6	10/29/2018	11/16/2018																																												
Release 2																																														
Release 3																																														
Areas	Teams																																													
▲ Fabrikam Fiber	Fabrikam Fiber Team																																													
Customer Service	Customer Service Team																																													
Phone	Fabrikam Fiber Team, Phone																																													
Voice	Voice																																													
Web	Fabrikam Fiber Team, Web																																													

Customize work-tracking processes

You and your teams can start using all work-tracking tools immediately after you create a project. But often, one or more users want to customize the experience to meet one or more business needs. Although you can customize the process easily through the user interface, you can establish a methodology for who manages the updates and evaluates requests.

NOTE

By default, users granted Stakeholder and higher access are granted permission to create, edit, and manage processes used to customize the work-tracking experience. If you want to lock down who can perform these tasks, set permissions at the organization level to **Deny**.

To learn more, see the following articles:

- [About process customization and inherited processes](#)
- [Customize a project](#)
- [Add and manage processes](#)

Customize work-tracking processes

You and your teams can start using all work-tracking tools immediately after you create a project. But often, one or more users want to customize the experience to meet one or more business needs. You can establish a

methodology for who manages the updates and evaluates requests.

To learn more, see [On-premises XML process model](#).

Review and update notifications

A number of notifications are predefined for each project you add. Notifications are based on subscription rules.

Subscriptions arise from the following areas:

- [Out-of-the-box or default subscriptions](#).
- [Team notifications](#), managed by a team administrator.
- Project notifications, managed by a member of the Project Administrators group.
- [Organization and collection level notifications](#), managed by a member of the Project Collection Administrators group.

If users believe they're getting too many notifications, they can [opt out of a subscription](#).

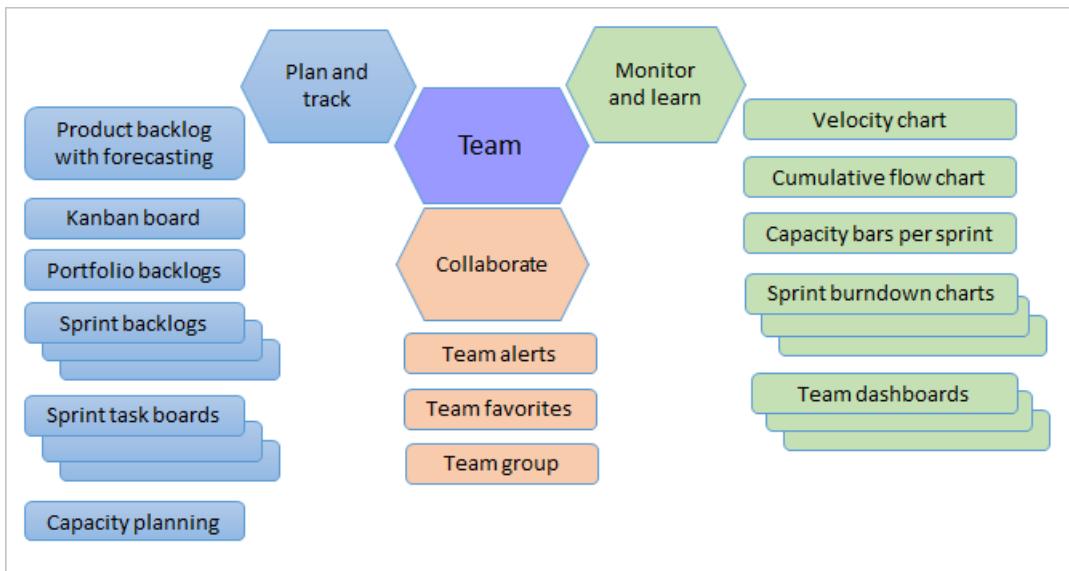
Description	Type	Notifies	State
Build			
 Build completes Notifies you when a build you queued or that was queued for you compl...	Build completed (any project)	 You	<input checked="" type="button"/> On
Code (Git)			
 Pull request reviewers added or removed Notifies you when you are added to a pull request or when a user is add...	Pull request (any project)	 You	<input checked="" type="button"/> On
 Pull request completion failures Notifies you when a pull request you created fails to complete	Pull request (any project)	 You	<input checked="" type="button"/> On
 Pull request changes Notifies you when changes are made to a pull request you created or are...	Pull request (any project)	 You	<input checked="" type="button"/> On
 A comment is left on a pull request Notifies you about comments made to a pull request you created or a di...	Pull request comment (any project)	 You	<input checked="" type="button"/> On

Configure an SMTP server

In order for team members to receive notifications, [you must configure an SMTP server](#).

Add teams to scale your organization

We recommend that you add teams as your organization grows. Each team gets [access to their own set of Agile tools](#) that they can customize.



To learn more, see the following articles:

- [About projects and scaling your organization](#)
- [Add a team, move from one default team to several teams](#)
- [Add a team administrator](#)

Install and manage extensions

An extension is an installable unit that adds new capabilities to your projects. You can find extensions in Azure DevOps to support the following functions:

- Planning and tracking of work items, sprints, scrums, etc.
- Build and release flows.
- Code testing and tracking.
- Collaboration among team members.

For example, to support [code search](#), install the [Code Search extension](#).

You want to tell your users about extensions and that they can [request an extension](#). To install and manage extensions, you must be an organization Owner, a member of the Project Collection Administrators group, or added to the [Manager role for extensions](#).

Set up billing

All organizations can add up to five users with Basic access and unlimited users with Stakeholder access. If you need to add more users or pay for additional services or extensions, [set up billing](#).

Next steps

[Manage projects](#)

Related articles

- [Security & identity](#)
- [Organization management](#)
- [About user, team, project, and organization-level settings](#)
- [Manage projects](#)
- [Security & identity](#)

- [Organization management](#)
- [About user, team, project, and organization-level settings](#)
- [TFS administration](#)

Add users to a project or team

8/1/2019 • 7 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this quickstart, you learn how to add users to a project or specific team. For anyone to access a project, they must be added to one of the default security groups or a custom group. Usually you add them to the Contributors group. For a quick look at what permissions are assigned to the default groups, see [Permissions and access](#).

The easiest way to add a number of users to a project is to add groups defined in [Azure Active Directory \(Azure AD\)](#) or [Active Directory \(AD\)](#).

IMPORTANT

If you're adding users to an organization in Azure DevOps and you don't use Azure AD, then you need to [add their "personal" Microsoft accounts to your account or project](#). After you've added them to one project, you can add them to additional projects using the procedures provided in this article.

Once users have been added to a project, you can browse for their display name or user name (email alias). Also, you can [add them to a specific team](#). To add a team, see [Add a team](#).

Prerequisites

- You must have a project. If you don't have a project yet, [create one](#).
- To add users to a project, you must be a member of the [Project Administrators group](#) or have your **Edit project-level information** set to **Allow**. You can add Stakeholders to the Project Administrators group and then they can add users to an organization or project.
- To add users to a team, you must be [added as a team administrator](#), or you must be a member of the Project Administrators Group, or have your **Edit project-level information** set to Allow.

Add users to a project

If you are adding a user to Azure DevOps for the first time, see [Add account users for Azure DevOps](#).

1. Open the web portal and choose the project where you want to add users or groups. To choose another project, see [Switch project, repository, team](#).
2. Choose **Project Settings** and then **Security**.

To see the full image, click to expand.

The screenshot shows the 'Project Settings' page for the 'Fabrikam Fiber' project. On the left, there's a sidebar with links like Overview, Work, Code, Build and release, and Packages. Below that is the 'Project settings' button, which is highlighted with a red box and a number '1'. To its right is the main content area. In the main content area, the 'Security' tab is selected, indicated by a red box and a number '2'. The 'General' section contains links for Overview, Services, Teams, Notifications, Service hooks, and Dashboards. Under the 'Teams' section, there's a 'Create group' button and a 'Filter users and groups' input field. Below that is a list of teams: Customer Service, Email, Fabrikam Fiber Team, Management team, Phone, Voice, and Web. There are also sections for 'Azure DevOps Groups' containing Build Administrators, Contributors, Deployment Group Administrators, Disallow access group, Endpoint Administrators, Endpoint Creators, Project Administrators, Project Collection Valid Users, and Security Service Group.

3. Under **Groups**, choose one of the following options:

- To add users who require read-only access to the project, choose **Readers**.
- To add users who contribute fully to this project or who have been granted Stakeholder access, choose **Contributors**.
- For users who need to administrate the project, choose **Project Administrators**. To learn more, see [Set permissions at the project-level or project collection-level](#).

4. Next, choose the **Members** tab.

Here we choose the **Contributors** group.

Screenshot of the 'Contributors' group page in Microsoft Teams. The left sidebar shows 'Teams' and 'Azure DevOps Groups'. The 'Contributors' group is selected and highlighted with a red box. The main area shows the 'Members' tab selected, with a table listing members and their details.

Display Name	Username Or Scope	
Customer Service	[Fabrikam Fiber]	Remove
Fabrikam Fiber Team	[Fabrikam Fiber]	
Management team	[Fabrikam Fiber]	
Phone	[Fabrikam Fiber]	
Voice	[Fabrikam Fiber]	
Web	[Fabrikam Fiber]	
Jia-hao Tseng	fabrikamfiber9@hotmail.com	

By default, the default team group and any other teams you add to the project, are included as members of the **Contributors** group. Add a new user as a member of a team instead, and the user automatically inherits Contributor permissions.

TIP

Managing users is much easier [using groups](#), not individual users.

5. Choose **+ Add** to add a user or a user group.
6. Enter the name of the user account into the text box. You can enter several identities into the text box, separated by commas. The system automatically searches for matches. choose the match(es) that meet your requirements.

Screenshot of the 'Add users and groups' dialog box. The input field contains 'Chris', and a search result for 'Christie Church' is displayed. The bottom right buttons are 'Save changes' and 'Cancel'.

NOTE

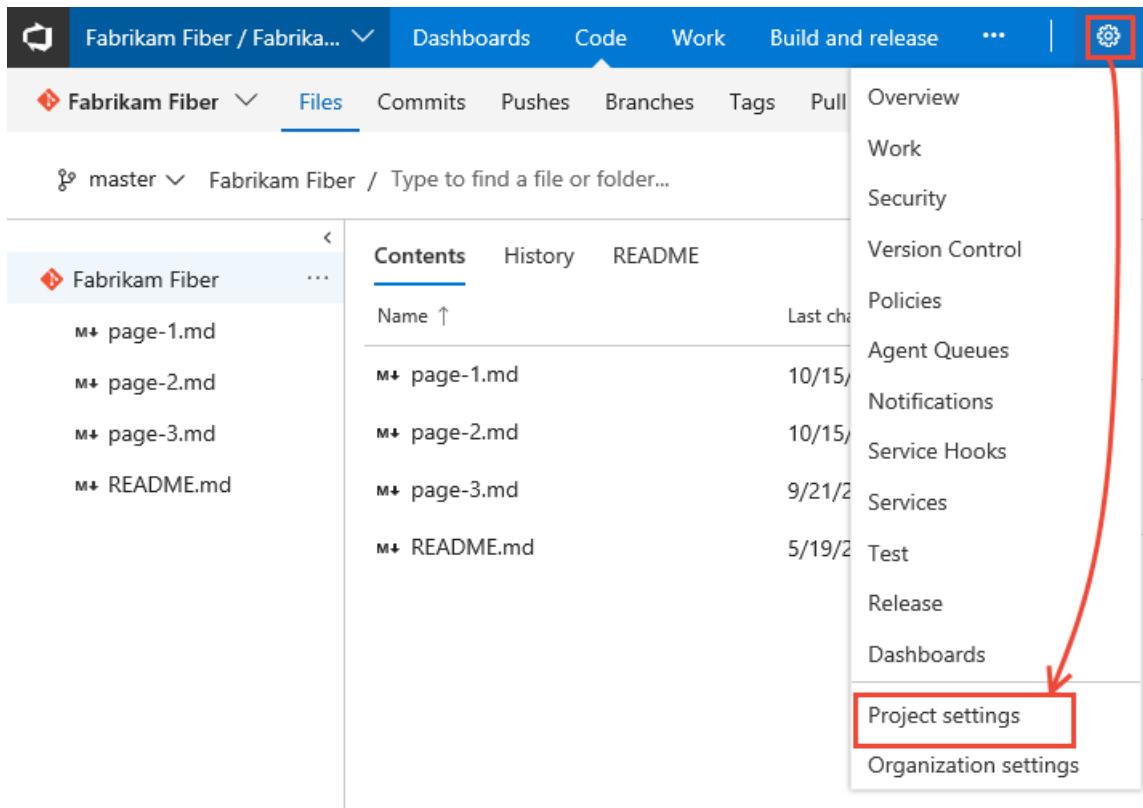
The first time you add a user or group to Azure DevOps, you can't browse to it or check the friendly name. After the identity has been added, you can just enter the friendly name.

7. In **Identities**, specify the name of the user or group you want to add.
8. Depending on the user, you may customize their permissions for other functionality in the project. For example, in [areas and iterations](#) or [shared queries](#).

NOTE

Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see [Permissions and access](#).

1. Open the web portal and choose the project where you want to add users or groups. To choose another project, see [Switch project, repository, team](#).
2. Choose the gear icon to open the administrative context.



3. Choose **Security** and under **Groups**, choose one of the following options:
 - To add users who require read-only access to the project, choose **Readers**.
 - To add users who contribute fully to this project, choose **Contributors**.
 - For users who need to administrate the project, choose **Project Administrators**. To learn more, see [Set permissions at the project-level or project collection-level](#).
4. Next, choose the **Members** tab.

Here we choose the Contributors group.

Create group

Filter users and groups

Teams

- Customer Service
- Fabrikam Fiber Team
- Management team
- Phone
- Voice
- Web

Azure DevOps Groups

- Build Administrators
- Contributors
- Project Administrators
- Project Valid Users
- Readers

Permissions Members Member of

+ Add... | Refresh | Search

Display Name	Username Or Scope	Remove
Customer Service	[Fabrikam Fiber]	
Fabrikam Fiber Team	[Fabrikam Fiber]	
Management team	[Fabrikam Fiber]	
Phone	[Fabrikam Fiber]	
Voice	[Fabrikam Fiber]	
Web	[Fabrikam Fiber]	
Jia-hao Tseng	fabrikamfiber9@hotmail.com	

TIP

Managing users is much easier [using groups](#), not individual users.

By default, the default team group and any other teams you add to the project, are included as members of the **Contributors** group. Add a new user as a member of a team instead, and the user automatically inherits Contributor permissions.

- Choose **+ Add** to add a user or a user group.
- Enter the name of the user account into the text box. You can enter several identities into the text box, separated by commas. The system automatically searches for matches.

Add users and groups

To add users or groups to this group, just type their sign-in addresses or group aliases

User or group: Chris

 Christie Church fabrikamfiber1@hotmail.com	<input type="checkbox"/>
---	--------------------------

Showing 1 result

Save changes Cancel

NOTE

The first time you add a user or group to Azure DevOps, you can't browse to it or check the friendly name. After the identity has been added, you can just enter the friendly name.

7. In **Identities**, specify the name of the user or group you want to add.
8. You may want to customize user permissions for other functionality within the project, such as [areas and iterations](#) or [shared queries](#).

NOTE

Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see [Permissions and access](#).

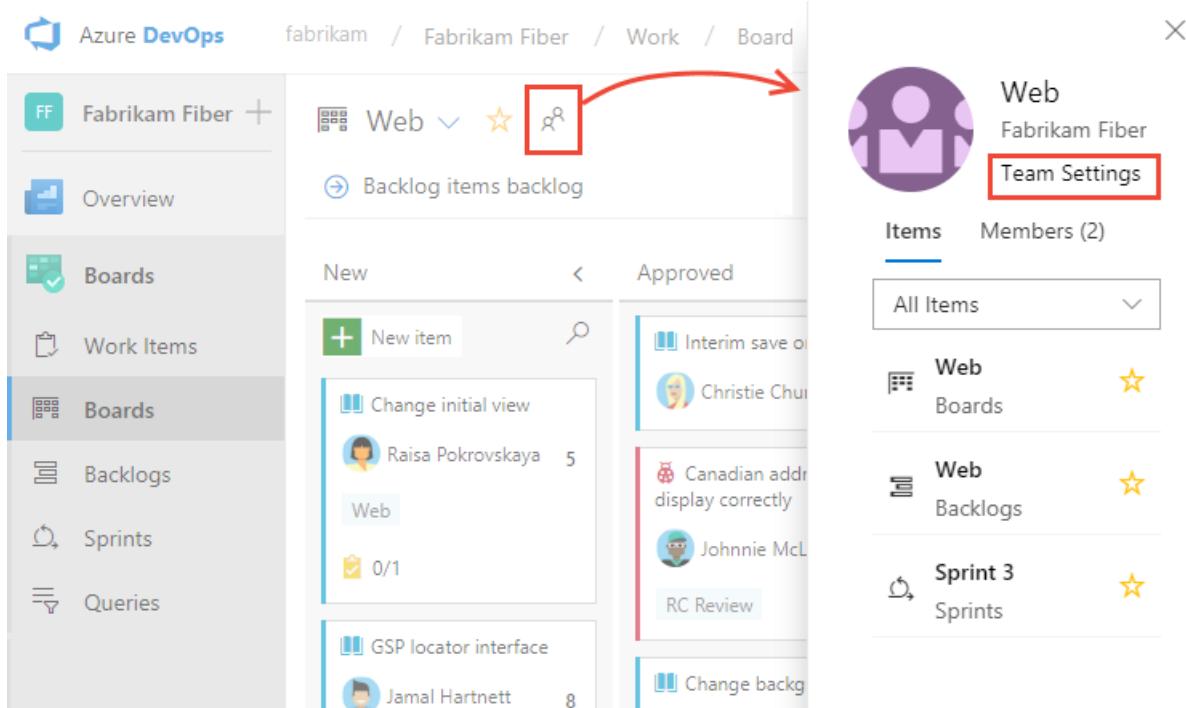
Add users to a team

Several Agile tools, like capacity planning, team alerts, and dashboard widgets are team-scoped. That is, they automatically reference the user accounts added as members of a team to support planning activities or sending alerts. To learn more, see [About teams and Agile tools](#).

You add team members from **Project Settings>Work>Team configuration**. You can quickly navigate to it from a team work tracking backlog, board, or dashboard.

1. Open a backlog or board for a team and choose the  team profile icon. Then choose **Team Settings**.

Here we open the Board for the Web team and from there the team profile.



The screenshot shows the Azure DevOps interface with the following details:

- Left Sidebar:** Shows navigation links for 'Overview', 'Boards', 'Work Items', 'Backlogs', 'Sprints', and 'Queries'.
- Breadcrumbs:** fabrikam / Fabrikam Fiber / Work / Board
- Current View:** Backlog items backlog for the 'Web' team.
- Team Profile Icon:** A purple icon representing the 'Web' team.
- Team Settings Dialog:**
 - Title:** Web Fabrikam Fiber
 - Buttons:** Team Settings (highlighted with a red box)
 - Items Tab:** Shows a dropdown for 'All Items' and a list of backlog items:
 - Interim save on
 - Canadian address display correctly
 - GSP locator interface
 - Change backg
 - Members Tab:** Shows 'Members (2)'.

2. If you need to switch the team context, use the team selector within the breadcrumbs.

Project Settings > Team configuration > Web

General Work General Iterations

- Overview
- Services
- Teams
- Security
- Notifications
- Service hooks

3. Choose **Add**.

Team Profile

Web

Members

+ Add... |

Display Name	Username Or Scope	Remove
Jamal Hartnett	fabrikamfiber4@hotmail.com	
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com	

Name
Web

Description
Enter a description

Administrators
Jamal Hartnett
Raisa Pokrovskaya
+ Add

4. Enter the sign-in addresses or display name for each account you want to add. Add them one at a time or all at the same time. You can enter several identities into the text box, separated by commas.

Add users and groups

To add users or groups to this group, just type their sign-in addresses or group aliases

User or group: Chris

Christie Church fabrikamfiber1@hotmail.com	
---	--

Showing 1 result

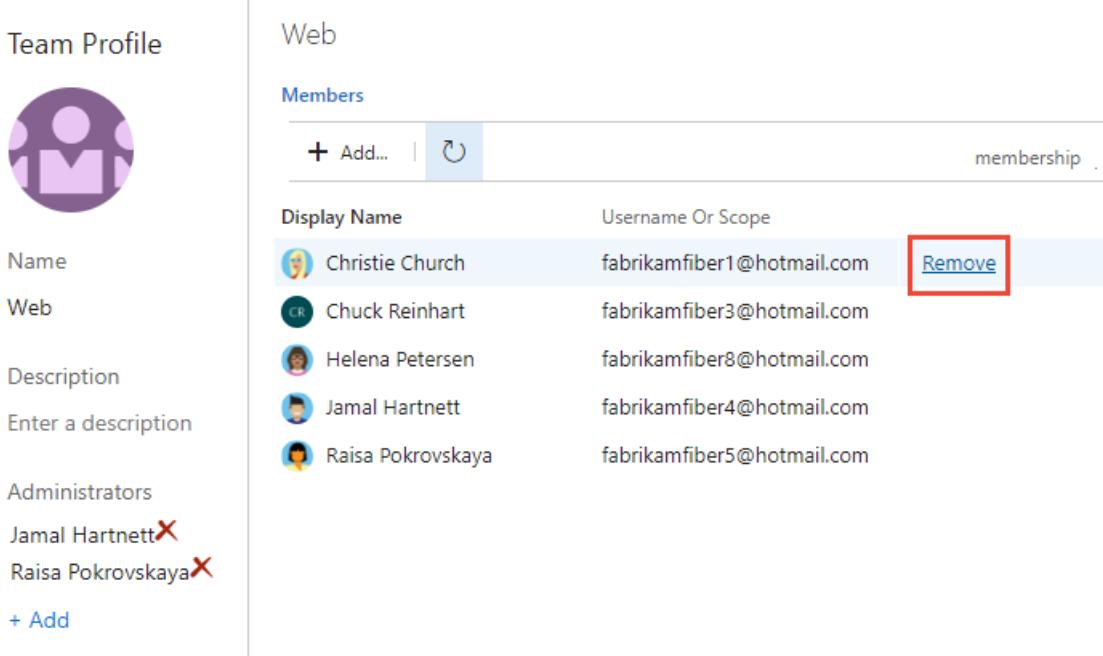
Save changes **Cancel**

TIP

You must enter user and group names one at a time. However, after entering a name, the account is added to the list, and you can enter another name in the Identities text box before choosing to save your changes.

You may need to choose the  refresh icon to see your updates.

5. To remove members, return to this page, highlight the user name and choose **Remove**.



The screenshot shows the 'Team Profile' page for a 'Web' team. On the left, there's a sidebar with 'Name' (Web), 'Description' (Enter a description), and 'Administrators' (Jamal Hartnett, Raisa Pokrovskaya, both with a red 'X'). The main area is titled 'Members' with a table:

Display Name	Username Or Scope	Action
Christie Church	fabrikamfiber1@hotmail.com	Remove
Chuck Reinhart	fabrikamfiber3@hotmail.com	
Helena Petersen	fabrikamfiber8@hotmail.com	
Jamal Hartnett	fabrikamfiber4@hotmail.com	
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com	

NOTE

To remove a team administrator as a team member, you must first remove them as an administrator.

6. To add an account as a team administrator, choose **Add** located in the Team Profile page. For details, see [Add a team administrator](#).
1. From the project admin context, open the **Overview** page, and then choose the team you want to add team members to.

Screenshot of the Project profile page in Microsoft Azure DevOps. The 'Overview' tab is selected. On the left, there's a sidebar with 'Name: Fabrikam Fiber', 'Process: MyScrum', and 'Description: Web, voice, and phone apps'. On the right, under 'Teams', there's a table showing five teams: Customer Service (7 members), Fabrikam Fiber Team (7 members, described as 'The default project team'), Management team (1 member), Phone (1 member, highlighted with a red box), and Voice (1 member). A 'New team' button and a refresh icon are also present.

2. Choose the **+ Add** to add a user or a user group.
3. Enter the sign-in addresses or display name for each user you want to add. Add them one at a time or all at the same time. You can enter several identities into the text box, separated by commas.

Screenshot of the 'Add users and groups' dialog box. It shows a search bar with 'Chris' typed in, a results list with 'Christie Church' and her email 'fabrikamfiber1@hotmail.com', and a note 'Showing 1 result'. At the bottom are 'Save changes' and 'Cancel' buttons.

TIP

You must enter user and group names one at a time. However, after entering a name, it is added to the list, and you can enter another name in the Identities text box before choosing to save your changes.

You may need to choose the  refresh icon to see your updates.

4. To remove members, return to this page, highlight the user name, and then choose **Remove**.

Team Profile



Phone

Members

+ Add... | ⚙

Display Name	Username Or Scope
Christie Church	fabrikamfiber1@hotmail.com
Chuck Reinhart	fabrikamfiber3@hotmail.com
Cristina Potra	fabrikamfiber6@hotmail.com
Jamal Hartnett	fabrikamfiber4@hotmail.com
Johnnie McLeod	fabrikamfiber2@hotmail.com
Raisa Pokrovskaya	fabrikamfiber5@hotmail.com
Cristina Potra	

NOTE

To remove a team administrator as a team member, you must first remove them as an administrator.

5. To add an account as a team administrator, choose **Add** located in the Team Profile page. For details, see [Add a team administrator](#).

Add users or groups to an access level

For on-premises deployments, you may need to set the access level for a user or group, particularly if those groups don't belong to the default access level. To learn more, see [Change access levels](#).

Add users or groups to SQL Server Reports

If your on-premises deployment is integrated with SQL Server Reports, you need to manage membership for those products separately from their websites. See [Grant permissions to view or create SQL Server reports in Azure DevOps](#).

Add users or groups to SharePoint or SQL Server Reports

If your on-premises deployment is integrated with a SharePoint product or SQL Server Reports, you need to manage membership for those products separately from their websites.

- [Set SharePoint site permissions](#)
- [Grant permissions to view or create SQL Server reports in Azure DevOps Server](#)

Next steps

[Add administrators or set permissions at the project or collection level](#)

Related articles

- To view permissions for yourself or another user, see [View permissions](#).
- [Set Git or TFVC repository permissions](#)
- [Set Git branch permissions](#)
- [Set build and release permissions](#)
- [Set permissions and access for work tracking](#)

Manage and configure team tools

9/4/2019 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015

In this article, learn how to configure team tools and manage teams in Azure DevOps.

Most permissions are governed by security groups or defined at the object level. Team settings are managed by the team administrator role. Users assigned as a team administrator can configure and manage all team tools. Specifically, when a team is added to a project, a project admin should [add one or more team administrators](#).

Then, those team admins should look at doing the following specific tasks:

- Add team members
- Configure area and iteration paths
- Configure backlogs and other common team settings
- Configure Kanban boards

Optional tasks to consider include:

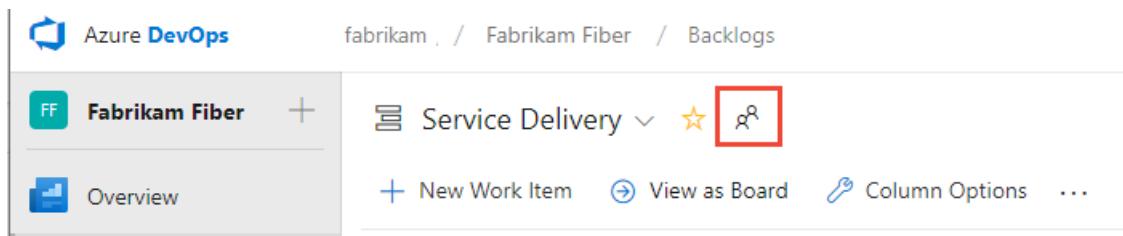
- Configure and manage team dashboards
- Configure team notifications

NOTE

In addition to team administrators, all members of the Project Administrators and Project Collection Administrators groups can manage settings for all teams. To add a team, see [Add teams](#).

Open the team profile and access team tools

- Open a team profile to quickly access items defined for a team. The team profile is available from the **Overview>Dashboards, Boards>Boards, Boards>Backlogs**, and **Boards>Sprints** pages.



A panel opens that shows all items defined for the team.

X

Service Delivery
Fabrikam Fiber
Team Settings

Items Members (7)

All Items ▾

	Service Delivery Boards	
	Service Delivery Backlogs	
	Sprint 2 Sprints	
	Overview Dashboards	

- You can filter the list to show only **Dashboards**, **Boards**, **Backlogs**, or **Sprints** by choosing from the menu.

Service Delivery
Fabrikam Fiber
Team Settings

Items Members (7)

All Items ▾

All Items

Dashboards

Boards

Backlogs

Sprints

- To view the team admins and members of the team, choose **Members**.

The screenshot shows a team settings interface. At the top left is a purple icon representing a team. To its right, the text "Service Delivery" and "Fabrikam Fiber" are displayed. Below this, there are links for "Team Settings" and "Members (7)". The "Members (7)" link is highlighted with a red box. Underneath these links, the word "Admins" is followed by a list item with a blue circular icon containing "CP" and the name "Cristina Potra". Below this, the word "Members" is followed by a list of six users, each with a small circular profile picture and their name: "Christie Church", "Chuck Reinhart", "Jamal Hartnett", "Johnnie McLeod", and "Raisa Pokrovskaya".

- To view or change the team configuration, choose **Team Settings**.

You can then complete the following tasks:

- Add [team members](#)
- Add [team admins](#)
- Navigate to [team notifications](#)
- Navigate to team [iterations](#) and [area paths](#).
- Update the [team description](#) or [profile picture](#).

Add users to a team

Several tools, such as capacity planning, team alerts, and dashboard widgets, are team-scoped. These tools automatically reference the users that are as members of a team to support planning activities or sending alerts.

To add users to a team, see [Add users to a project or specific team](#).

All members of a team can favorite team artifacts and define work item templates. For details, see:

- [Set personal or team favorites](#)
- [Use templates to add and update work items](#).

If team members don't have access to all the features they want, check that they have [the permissions needed for those features](#).

Configure team areas and iterations

Many Agile tools depend on the area and iteration paths that are configured for the team. To learn more about configuring team areas and iterations, see [About teams and Agile tools](#).

Once project administrators have [added Area Paths](#) and [Iteration Paths](#) for a project, team administrators can select the area and iteration paths associated with their team. These settings affect a number of Agile tools available to the team.

These include making the following associations for each team:

- **Select team area paths**

Can select the default area path(s) associated with the team. These settings affect a number of Agile tools available to the team.

- **Select team iteration paths or sprints** Can select the default area path(s) associated with the team. These settings affect a number of Agile tools available to the team.

To learn more, see [Define area paths and assign to a team](#) and [Define iteration paths and configure team iterations](#).

Configure team backlogs and other common settings

Team administrators can choose which backlog levels are active for a team. For example, a feature team may choose to show only the product backlog and a management team may choose to show only the feature and epic backlogs. Also, admins can choose whether bugs are treated similar to user stories and requirements or as tasks.

Team admins can also choose which days are non-working days for the team. Sprint planning and tracking tools automatically consider days off when calculating capacity and sprint burndown.

You can configure most of your team settings from the common configuration dialog.

NOTE

The common configuration Settings dialog is available for TFS 2015.1 and later versions.

NOTE

To understand the differences between backlogs, boards, taskboards, and Delivery plans, see [Backlogs, boards, and plans](#). If your backlog or board doesn't show the work items that you expect or want, see [Set up your backlogs and boards](#).

NOTE

To understand the differences between backlogs, boards, and taskboards, see [Backlogs, and boards](#). If your backlog or board doesn't show the work items that you expect or want, see [Set up your backlogs and boards](#).

1. (1) Check that you selected the right project, (2) choose **Boards > Boards**, and then (3) select the correct team from the team selector menu.

The screenshot shows the Azure DevOps Boards page. At the top left is the Azure DevOps logo and the product name "fabrikam / Fabrikam Fiber". A red box highlights the breadcrumb path "fabrikam / Fabrikam Fiber". At the top right is a red circle with the number "1". Below the header is a navigation bar with items: "Fabrikam Fiber" (with a plus sign), "Fabrikam Fiber Team" (highlighted with a red box), "Backlog items" (with a red star icon), and "Backlog items" again (with a red circle containing the number "3"). The main area is divided into three columns: "Backlog", "Analyze", and "Develop". The "Backlog" column contains a "New item" button and a card for "Add an information form" by "Raisa Pokrovskaya" in "Iteration ... Sprint 3". The "Analyze" column contains a card for "Welcome back page" by "Johnnie McLeod" in "Iteration ... Sprint 3". The "Develop" column contains a card for "Slow form" by "Jam" with a progress bar showing "0/2". On the left sidebar, under the "Boards" heading (highlighted with a red box and a red circle containing "2"), there are links for "Overview", "Work Items", "Boards", "Backlogs", and "Sprints".

2. Make sure that you select the team backlog or board that you want to configure using the team selector. To learn more, see [Use breadcrumbs and selectors to navigate and open artifacts](#).
3. Choose the product or portfolio backlog from the board-selection menu.

The screenshot shows the "Backlog items backlog" page. At the top left are icons for "Web", "Backlog items backlog", and a gear icon. At the top right are icons for "Backlog items" (highlighted with a red box), "Epics", "Features", and "Backlog items". A red box highlights the "Backlog items" dropdown menu. The main area displays a cumulative flow chart with two bars and a gear icon. The bottom right corner of the gear icon is highlighted with a red box.

4. Choose the gear icon to configure the board and set general team settings.

The screenshot shows the same "Backlog items backlog" page as the previous one, but the gear icon at the bottom right is now highlighted with a red box.

5. Choose a tab under any of the sections—**Cards**, **Board**, **Charts**, and **General**—to configure the cards or boards, the cumulative flow chart, or other team settings.

Settings

Cards

- Fields**
- Styles
- Tag colors

Board

- Columns
- Swimlanes
- Card reordering

Charts

- Cumulative flow

General

- Backlogs
- Working days
- Working with bugs

Fields

Show the important information to your team. Fields are editable directly on the card.

User Story
Bug

Core fields

- Show ID
- Show Assigned To as:

Avatar and full name (default)

▼

- Show Story Points
- Show Tags

Additional fields

Add up to 10 fields in the order that you want them to appear on the card.

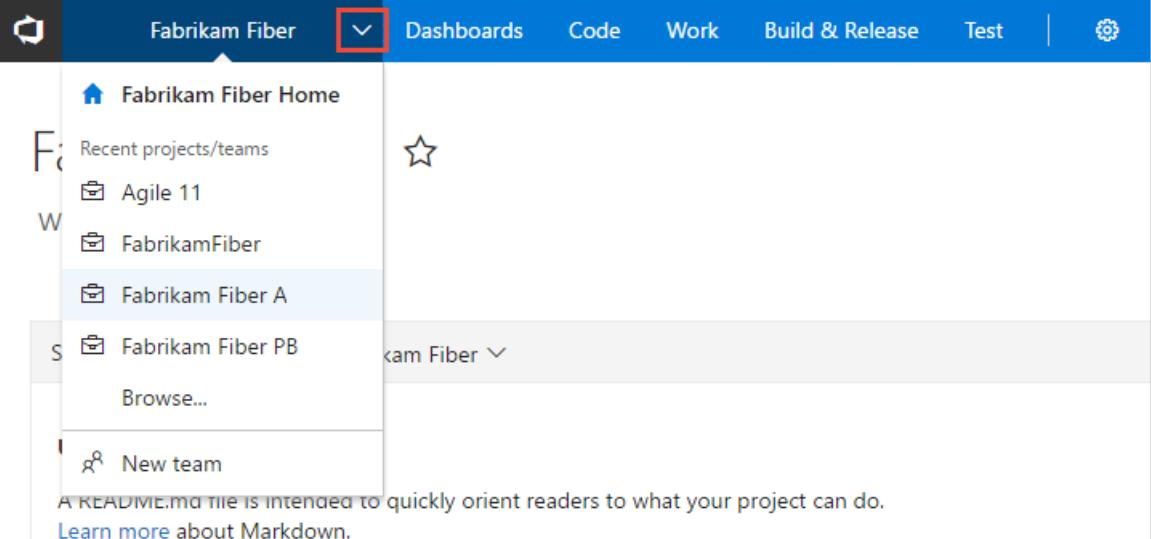
+ Field

Show empty fields

- Check if you want to display fields, even when they are empty.

Save
Cancel

1. Make sure that you select the team from the project/team selector. You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose **Browse...** or choose the  Azure DevOps logo to [access the Projects page](#).



The screenshot shows the Azure DevOps navigation bar. The 'Fabrikam Fiber' team is selected, indicated by a red box around the dropdown menu icon and the team name. Other options in the dropdown include 'Recent projects/teams', 'Agile 11', 'FabrikamFiber', 'Fabrikam Fiber A', 'Fabrikam Fiber PB', 'Browse...', and 'New team'. The main dashboard area shows a star icon and some placeholder text about a README.md file.

2. Open **Work>Backlogs>Board**.

3. Choose the board you want to configure and then choose the gear icon to configure the board and set general team settings.

For example, from the Kanban board ...

4. Choose a tab under **Cards** or **Board** to configure the cards and Kanban board columns and swimlanes.

![Common configuration dialog team settings](./boards/boards/_img/customize-cards/common-config-141.png)

1. Make sure that you select the team from the project/team selector. You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose **Browse...** or choose the Azure DevOps logo to [access the Projects page](#).

2. Open **Work>Backlogs>Board**.

Column	Status	Count
Backlog	Active	6/5
Active	Resolved	6/5

3. Choose the board you want to configure and then choose the gear icon to configure the board and set general team settings.

For example, from the Kanban board ...

Column	Status	Count
Backlog		
Analyze		4/10
Develop		1/5
Test		4/5

4. Choose a tab under **Cards** or **Board** to configure the cards and Kanban board columns and swimlanes.

Settings

Cards

Fields

Styles

Tag colors

Board

Columns

Swimlanes

Card reordering

Charts

Cumulative flow

General

Backlogs

Working days

Working with bugs

Fields

Show the important information to your team. Fields are editable directly on the card.

User Story **Bug**

Core fields

Show ID

Show Assigned To as:

Avatar and full name (default) ▾

Show Story Points

Show Tags

Additional fields

Add up to 10 fields in the order that you want them to appear on the card.

+ Field

Show empty fields

Check if you want to display fields, even when they are empty.

Save **Cancel**

For details on each configuration option, see one of the following articles:

AREA	CONFIGURATION TASK
Cards	<ul style="list-style-type: none">• Add fields• Define styles• Add tag colors• Enable annotations• Configure inline tests
Boards	<ul style="list-style-type: none">• Add columns• Add swimlanes• Card reordering• Configure status badges
Chart	<ul style="list-style-type: none">• Configure cumulative flow chart
General	<ul style="list-style-type: none">• Backlogs• Working days• Working with bugs
AREA	CONFIGURATION TASK

Cards	<ul style="list-style-type: none"> • Add fields • Define styles • Add tag colors • Enable annotations • Configure inline tests
Boards	<ul style="list-style-type: none"> • Add columns • Add swimlanes • Card reordering
Chart	<ul style="list-style-type: none"> • Configure cumulative flow chart
General	<ul style="list-style-type: none"> • Backlogs • Working days • Working with bugs

For details on each configuration option, see one of the following articles:

AREA	CONFIGURATION TASK
Cards	<ul style="list-style-type: none"> • Add fields • Define styles • Add tag colors
Boards	<ul style="list-style-type: none"> • Add columns
Chart	<ul style="list-style-type: none"> • Configure cumulative flow chart
General	<ul style="list-style-type: none"> • Backlogs • Working days • Working with bugs

Configure Kanban boards

Team administrators can fully customize the team's Kanban boards associate with the product and portfolio backlogs. You configure a Kanban board by first defining the columns and WIP limits from the common configuration dialog. For guidance, see [Kanban basics](#).

- [Columns](#)
- [WIP limits](#)
- [Definition of Done](#)

Additional elements you can configure include:

- [Split columns](#)
- [Swimlanes](#)
- [Card fields, styles, tag colors, annotations, and card reordering](#)

Configure sprint Taskboards

Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards as well as addition of customized columns. For details, see [Customize sprint Taskboards](#).

Similar to Kanban boards, each sprint Taskboard can be customized to support information-rich, color-coded cards. For details, see [Customize sprint Taskboards](#).

Add and manage team dashboards

By default, all team members can add and edit team dashboards. In addition, team administrators can manage permissions for team dashboards. For details, see [Add and manage dashboards](#).

Team administrators can add, configure, and manage permissions for team dashboards. For details, see [Add and manage dashboards](#).

Update team description and picture

Team settings also include the team name, description, and team profile image. To add a team picture. Open the Team Profile and choose the picture icon. The maximum file size is 4 MB.

Manage team notifications

Team administrators can add and modify alerts so that the team can receive email notifications as changes occur to work items, code reviews, source control files, and builds. A number of alerts are defined for each team. For details, see [Manage team alerts](#).

Manage team rooms

Team administrators can add users and events to team rooms, and add team rooms. Team rooms are chat rooms limited to team members. For details, see [Collaborate in a team room](#).

NOTE

Team rooms are deprecated for TFS 2018 and later versions as described in [Deprecation of team rooms](#) blog post. Several good solutions are available that integrate well with TFS that support notifications and chat, such as [Microsoft Teams](#) and [Slack](#).

Related articles

- [About projects and scaling your organization](#)
- [About teams and Agile tools](#)
- [Add teams](#)
- [Add a team administrator](#)

Tutorial: Set personal or team favorites

8/1/2019 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Favorite  those views that you frequently access. You can favorite all sorts of Azure DevOps features and tools—such as a project, repository, build pipeline, dashboard, backlog, board, or query. You can set favorites for yourself or your team.

As your code base, work tracking efforts, developer operations, and organization grows, you'll want to be able to quickly navigate to those view of interest to you and your team. Setting favorites allows you to do just that.

Team favorites are a quick way for members of your team to quickly access shared resources of interest. You favorite an item for yourself by choosing the  star icon. The favorited item will then show up easily from one or more directory lists. You set favorites for a team through the context menu for the definition, view, or artifact.

In this tutorial you'll learn how to view your personal favorites and to favorite or unfavorite the following views:

- Project or team
- Dashboard
- Team backlog, board, shared query, or other Azure Boards view
- Repository
- Build and release definition
- Test plans

- Project
- Shared query
- Repository
- Build and release definition
- Test plans

Prerequisites

- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.
- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.

- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.

- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.

For details about the different access levels, see [About access levels](#).

View personal favorites

Access views that you have favorited by choosing the inbox icon, and then choosing **Favorites**.

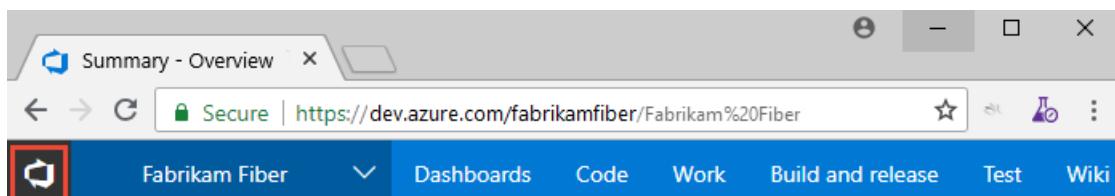
The screenshot shows the 'Favorites' section of the Azure DevOps interface. At the top, there's a navigation bar with icons for search, inbox (highlighted with a red box), library, and profile. Below the navigation bar, there are tabs for 'Work Items', 'Pull requests', and 'Favorites' (also highlighted with a red box). The main content area is organized into sections: 'Projects', 'Teams', 'Dashboards', 'Plans', and 'Queries'. Each section lists items with a star icon indicating they are favorited. For example, under 'Teams', there are three items: 'Phone', 'Voice', and 'Web', each with a yellow star. Under 'Plans', there are two items: 'Backlog team plans' and 'Fabrikam Fiber Feature plans', each with a yellow star. Under 'Queries', there are three items: 'All items', 'All items on all projects', and 'Assigned to me', each with a yellow star.

Category	Item	Status
Projects	Fabrikam Fiber	★
Teams	Phone	★
	Voice	★
	Web	★
Dashboards	Fabrikam Fiber Team Analytics	★
Plans	Backlog team plans	★
	Fabrikam Fiber Feature plans	★
Queries	All items	★
	All items on all projects	★
	Assigned to me	★

NOTE

If a service is disabled, then you can't favorite an artifact or view of that service. For example, if **Boards** is disabled, then the favorite groups—Plans, Boards, Backlogs, Analytics views, Sprints, and Queries and all Analytics widgets—are disabled. To re-enable a service, see [Turn an Azure DevOps service on or off](#).

1. Access views that you have favorited by choosing the Azure DevOps logo to open **Projects**.



2. Choose **My Favorites** to quickly access any view or item that you've marked as a favorite.

Favorites

Filter favorites



Queries

Bug Triage	Fabrikam Fiber	.../Shared Queries/Current Iteration	
My Bugs	Contoso	Shared Queries	
Open User Stories	Contoso	.../Shared Queries/Current Iteration	
Product Planning	Fabrikam Fiber	Shared Queries	
Product Planning	Contoso	Shared Queries	

Favorite a project or team

1. To favorite a project, open the project **Summary** page and choose the star icon.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, a sidebar menu has 'Summary' selected and highlighted with a red box. The main content area displays the project's logo ('FF'), its name ('Fabrikam Fiber'), and a yellow star icon indicating it is favorited. Below this, there is a section for 'Web, voice, and phone apps'. A 'README.md' file is listed with the following content:

```
minor modification to test development section in mobile form  
Update this README.md file.  
A README.md file is intended to quickly orient readers to what your project
```

2. To favorite a team artifact, open **Boards>Boards** or **Boards>Backlogs**. Select the team you want to favorite from the team selector and choose the star icon.

The screenshot shows the 'Boards > Backlog items backlog' page. The 'Phone' team icon is selected, indicated by a blue arrow pointing to it. The star icon next to the team name is highlighted with a red box. The backlog items section shows a single item: 'Backlog items backlog'.

3. To favorite other team artifacts, choose the team icon, and then choose the star icon next to one of the listed artifacts.

The screenshot shows the 'Phone' project settings page. At the top, there's a purple circular icon with three stylized human figures. Below it, the project name 'Phone' and the team name 'Fabrikam Fiber' are displayed, along with a 'Team Settings' link. A navigation bar at the bottom has 'Items' (underlined) and 'Members (1)' options. A dropdown menu shows 'All Items'. Below this, three items are listed: 'Phone Boards' (with a board icon), 'Phone Backlogs' (with a document icon), and 'Phone Sprints' (with a circular arrow icon). Each item has a yellow star icon to its right.

Favorite a project

To favorite a project, open the project **Summary** page and choose the star icon.

The screenshot shows the 'FabrikamFiber' project summary page. At the top, there's a blue header with the project name, a search bar, and other navigation links. The main area features a large card for 'FabrikamFiber' with a yellow star icon. Below it, a callout box highlights 'Continuous integration' with a 'Setup Build' button. To the right, there's a sidebar with sections for 'Members' (listing 'K' and a plus sign), 'Activity' (empty), 'Code' (empty), and 'Build & Rel' (empty).

Or, you can favorite a project from the **Projects** page by choosing the star icon next to the project.

Favorite a dashboard

- From **Overview>Dashboards**, open the selector and choose the **Browse all dashboards** option.

The screenshot shows the Microsoft Power BI interface with the title 'Fabrikam Team Overview'. The left sidebar contains a search bar and sections for 'Favorites', 'Account Management', 'Customer Profile', and 'Fabrikam Team'. A red box highlights the 'Browse all dashboards' button at the bottom. On the right, there are several cards: one green card showing '6 items', another purple card showing '0 Commits in last 7 d...', and a third card partially visible at the bottom labeled 'All items by State'.

2. The **Mine** page shows your favorited dashboards, and all dashboards of teams that you belong to. The **All** page (shown below) lists all dashboards defined for the project in alphabetical order. You can filter the list by team or by keyword.

Dashboards

Mine **All** | + New dashboard 

Filter dashboards Filter by team ▾

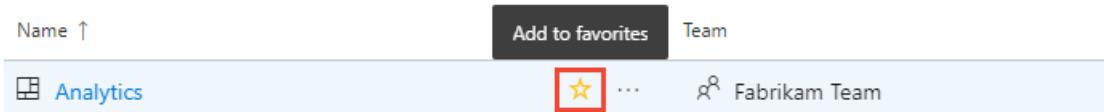
Name ↑	Team
Analytics	Fabrikam Team
Bug status	Fabrikam Team
Bugs	Internet
Overview	Account Management
Overview	Customer Profile
Overview	Email
Overview	Fabrikam Team
Overview	Internet
Overview	Phone
Overview	Service Delivery
Overview	Service Status
Team Guidance	Fabrikam Team
Work in Progress	Internet

Search
 Account Management
 Customer Profile
 Email
 Fabrikam Team
 Internet
 Phone
 Service Delivery
 Service Status


TIP

You can change the sort order of the list by choosing the column label.

3. To favorite a dashboard, hover over the dashboard and choose the star icon.



Favoriting a dashboard will cause it to appear on your **Favorites** page and towards the top in the **Dashboards** selection menu.

Favorite a team's backlog, Kanban board, or other view

You can favorite several Agile tools for a team from a **Boards** page.

1. Choose **Boards**, and then choose the page of interest, such as **Boards**, **Backlogs**, or **Sprints**.

For example, here we choose (1) **Work** and then (2) **Backlogs**.

A screenshot of the Boards page in Azure DevOps. On the left, there's a sidebar with 'Fabrikam Fiber' and a list of sections: Overview, Boards, Work Items, Boards, Backlogs (which is highlighted with a red box), Sprints, and Queries. The main area shows 'Fabrikam Fiber Team' with a star icon. It has buttons for 'New Work Item', 'Backlog items Board', and 'Backlog items'. Below is a table of backlog items:

Order	Assigned To	State	Title
1	Jamal Hartnett	Committed	> Hello World Web Site
2	Jamal Hartnett	Committed	> Slow response on informa...
3	Raisa Pokrovskaya	New	> Add an information form
4	Raisa Pokrovskaya	New	> Change initial view
5	Christie Church	Committed	> Secure sign-in
6	Johnnie McLeod	Approved	> Welcome back page
7	Christie Church	Committed	> Cancel order form

To choose a specific team backlog, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

Fabrikam Fiber Team

Search team backlogs

My Team Backlogs

- Account Management
- Customer Profile
- Fabrikam Team
- Phone
- Service Delivery
- Service Status
- Shopping Cart

Browse all backlogs

- Choose the star icon to favorite a team backlog. Favorited artifacts (favorited icon) appear on your **Favorites** page and towards the top of the team backlog selector menu.

Favorite a shared query

Open **Boards>Queries** and choose the **All** page. Expand a folder as needed. Choose the star icon next to the query you want to favorite.

Or, open the context menu of the query, and then select **Add to Team Favorites**, and then select from the list of teams.

Queries

Favorites All + New query Filter by keywords

Title

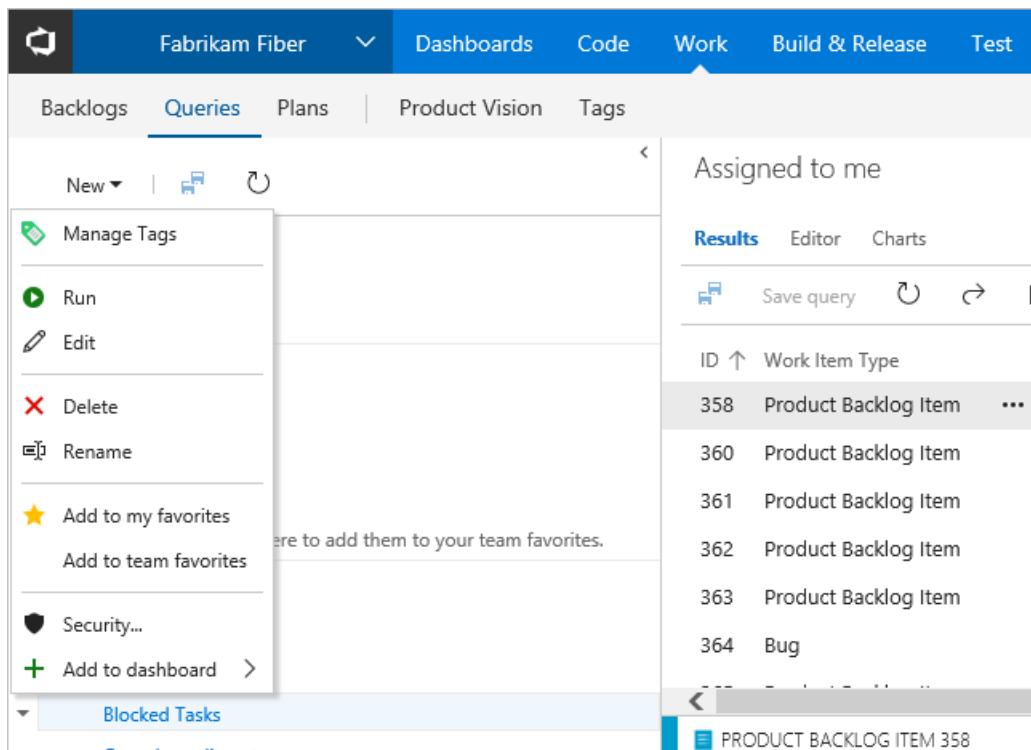
- > My Queries
- ▽ Shared Queries
 - Current Sprint
 - Blocked Tasks
 - Open Impediments
 - Test Cases
 - Unfinished Work
 - Work in Progress
 - Triage folder
 - All items
 - All items in a tree query
 - Feedback

Run query Edit Rename Delete Add to Team Favorites Security... Manage Tags

Customer Service
Fabrikam Fiber Team
Management team
Phone

You can also set a query as a personal favorite by opening the query and choosing the star icon.

Open **Work>Queries**. Next, open the *** actions icon menu of the shared query you want to favorite, and then select **Add to my favorites** or **Add to team favorites**.



The screenshot shows the 'Queries' page in the Azure DevOps interface. A context menu is open over a query result. The menu includes options like 'Manage Tags', 'Run', 'Edit', 'Delete', 'Rename', 'Add to my favorites' (which is highlighted with a yellow star icon), 'Add to team favorites', 'Security...', and 'Add to dashboard'. The main pane displays a list of results for the query 'Assigned to me', with one item selected: '358 Product Backlog Item'.

Favorite a delivery plan

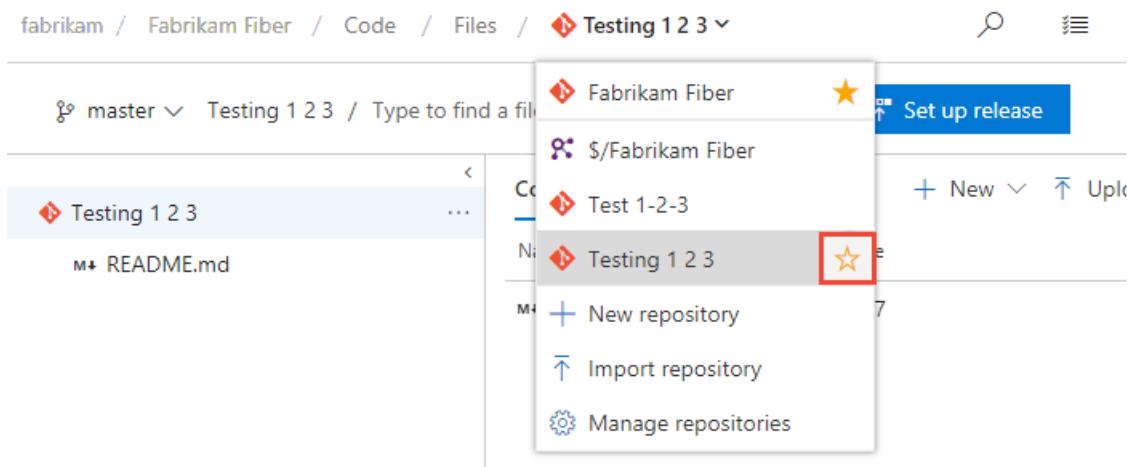
To learn more about delivery plans, see [Review team Delivery Plans](#).

To mark a delivery plan as a favorite, open the **Boards>Plans** page and choose the  star icon next to the Delivery Plan.

To mark a delivery plan as a favorite, open the **Work>Plans** page and choose the  star icon next to the Delivery Plan.

Favorite a repository

From any **Repos** page, open the repository selector and choose the  star icon for the repository you want to favorite.



The screenshot shows the 'Code' page in the Azure DevOps interface. A context menu is open over the repository 'Testing 1 2 3'. The menu includes options like 'Fabrikam Fiber', '\$/Fabrikam Fiber', 'Test 1-2-3', 'Testing 1 2 3' (which is highlighted with a yellow star icon), 'New repository', 'Import repository', and 'Manage repositories'. The repository 'Testing 1 2 3' is currently selected in the list.

From any **Code** page, open the repository selector and choose the star icon next to the repository you want to favorite.

The screenshot shows the 'Files' tab selected in the top navigation bar. On the left, there's a sidebar with 'Favorites' and a 'Filter repositories' search bar. The main area displays a list of repositories under 'All repositories'. The 'Fabrikam Fiber' repository is highlighted with a red box around its star icon in the favorites section. To the right, there's a detailed view of the repository's contents, including files like 'page-1.md', 'page-2.md', 'page-3.md', and 'README.md' with their respective last change dates and commit IDs.

Favorite a build pipeline

Open **Pipelines>Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to my favorites** or **Add to team favorites**.

The screenshot shows the 'Build Definitions' page with tabs for 'Mine', 'Definitions', 'Queued', and 'XAML'. It includes a search bar and buttons for '+ New' and '+ Import'. Below, a list of builds is shown, with the first one, 'fabrikam build', having a context menu open. The menu items include 'Queue new build...', 'Edit definition', 'Pause', 'View builds', 'Add to my favorites' (which is highlighted with a red box), 'Add to team favorites >', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', 'Security...', and '+ Add to dashboard >'.

Open **Build and Release>Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to**

[my favorites](#) or [Add to team favorites](#).

The screenshot shows the 'Build Definitions' page in the Azure DevOps interface. At the top, there are tabs for 'Mine', 'All Definitions', 'Queued', and 'XAML'. A search bar at the top right contains the placeholder 'Build ID or build number' with a magnifying glass icon. Below the tabs, columns for 'Recently built', 'Status', and 'Triggered by' are shown. A specific build definition, 'fabrikam build', is selected, indicated by a checkmark icon and a star icon. A context menu is open next to the build name, listing options: 'Queue new build...', 'Edit...', 'View definition summary', 'Add to my favorites' (which is highlighted with a red box), 'Add to team favorites', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', and 'Security...'. The status column for the selected build shows 'No builds have r...'.

Favorite a test plan

To learn more about test plans, see [Create a test plan and test suite](#).

To mark a test plan as a favorite, open **Test Plans>Test Plans** and choose the star icon next to a test plan from the menu that shows All test plans.

To mark a test plan as a favorite, open the **Test>Test Plans** page and choose the star icon next to a test plan from the menu that shows All test plans.

Unfavorite a view you've favorited

You can unfavorite an artifact from your **Favorites** page. Choose the inbox icon, and then choose **Favorites**. Choose the favorited icon of a currently favorited artifact.

The screenshot shows the Microsoft Teams ribbon bar. The tabs visible are Work Items, Pull requests, and Favorites. The Favorites tab is highlighted with a red box. Other icons in the ribbon include a search icon, a gear icon, and a user profile icon.

Projects

- Fabrikam Fiber

Teams

- Phone
 Voice
 Web

Dashboards

- Fabrikam Fiber Team Analytics

Plans

- Backlog team plans
 Fabrikam Fiber Feature plans

Queries

- All items
 All items on all projects
 Assigned to me

Similarly, you can unfavorite an artifact from the same page where you favorited it.

You can unfavorite an artifact from the **Projects>Favorites** page and choose the favorited icon of a currently favorited artifact.

Similarly, you can unfavorite an artifact from the same page where you favorited it.

Try this next

[Follow a user story, bug, issue, or other work item or pull request](#)

Related articles

- [Manage personal notifications](#)
- [Set your preferences](#)

Tutorial: Follow a user story, bug, issue, or other work item or pull request

8/1/2019 • 4 minutes to read • [Edit Online](#)

[Azure Boards](#) | [Azure Repos](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#)

To get notified of changes made to a work item or a pull request, you can elect to follow them.

This article shows you how to:

- Follow a work item
- Follow a pull request
- Manage work items that you're following

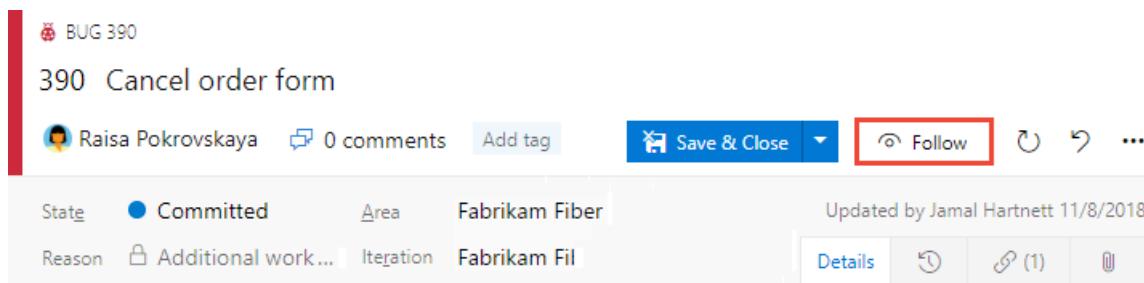
You must configure an [SMTP server](#) in order for team members to receive notifications.

Prerequisites

- You must connect to a project. If you don't have a project yet, [create one](#).
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, [Add users to a project or team](#).
- To view or follow work items, you must be granted **Stakeholder** access or higher. For details, see [About access levels](#). Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see [Set permissions and access for work tracking](#).
- To view or follow pull requests, you must have **Basic** access or higher.
- You must connect to a project. If you don't have a project yet, [create one](#).
- You must be added to a project as a member of the **Contributors** or **Project Administrators** security group. To get added, [Add users to a project or team](#).
- To view or follow work items, you must be granted **Stakeholder** access or higher. For details, see [About access levels](#). Also, you must have your **View work items in this node** and **Edit work items in this node** permissions set to **Allow**. By default, the **Contributors** group has this permission set. To learn more, see [Set permissions and access for work tracking](#).
- To view or follow pull requests, you must have **Basic** access or higher.

Follow a work item

When you want to track the progress of a single work item, choose the  [Follow](#) follow icon. This signals the system to notify you when changes are made to the work item.



The screenshot shows a work item details page for a bug titled "BUG 390". The work item ID is 390, and the title is "Cancel order form". The state is "Committed". The area is "Fabrikam Fiber", and the iteration is "Fabrikam Fil". The work item was updated by "Jamal Hartnett" on 11/8/2018. There are 0 comments, and the "Follow" button is highlighted with a red border. Below the main details, there are buttons for "Details", "Edit", and "Delete".

NOTE

The **Follow a work item** feature is available from TFS 2017 and later versions. The **Follow a pull request** feature is available from TFS 2017.1 and later versions. To update your on-premises TFS, visit the [Visual Studio downloads page for Team Foundation Server](#).

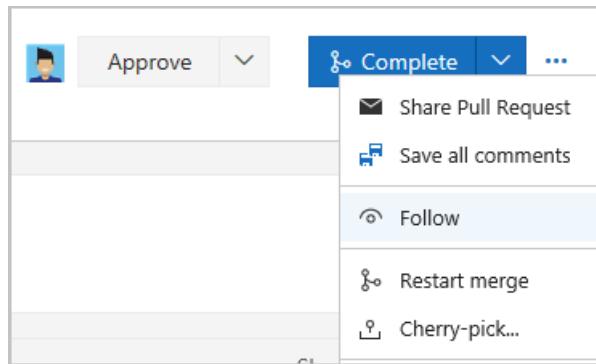
You'll only receive notifications when other members of your team modifies the work item, such as adding to the discussion, changing a field value, or adding an attachment.

Notifications are sent to your preferred email address, which [you can change from your user profile](#).

To stop following changes, choose the  [Following](#) following icon.

Follow a pull request

To track the progress of a single pull request, choose the  actions icon for the pull request, and select the  [Follow](#) option. This signals the system to notify you when changes are made to the PR.



You'll only receive notifications when other members of your team modifies the PR, such as adding to the discussion or adding an attachment.

Notifications are sent to your preferred email address, which [you can change from your user profile](#).

To stop following changes, open the PR context menu and choose the  [Following](#) following icon.

Manage work items that you're following

You can review and manage all the work items you've selected to follow.

Open **Boards>Queries**, choose **All**, and under **My Queries**, choose **Followed work items**.

Favorites **All** | + New query

Title

My Queries

- Active bugs
- All Items
- Assigned to me
- Closed bugs
- Followed work items**
- Fabrikam Fiber Team - Backlog items
- Following - my query

From this view, you can view all items you're following across all projects. Also, you can perform similar actions supported with a query results view, such as:

- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from **Boards>Work Items** and pivot to **Following**.

Following | + New Work Item | Open in Queries | Column Op

ID	Assigned To	State	Title
375	Jamal Hartnett	● Committed	Check service status
361	Christie Church	● Approved	Interim save on long form
384	Christie Church	● Committed	Secure sign-in
360	Raisa Pokrovskaya	● New	Change initial view
436	Jamal Hartnett	● Committed	Hello World Web Site

Open **Work>Queries** and choose **Followed work items**.

The screenshot shows the Microsoft Team Services interface for the 'Fabrikam Fiber' project. The top navigation bar includes links for HOME, CODE, WORK (which is selected), BUILD, and TEST. Below the navigation is a 'Queries' section with tabs for Backlogs and Queries. A sidebar on the left contains sections for 'Assigned to me', 'Followed work items' (which is highlighted with a red box), 'Unsaved work items', 'My favorites' (with a note to drag queries here), 'Team favorites' (with a note to drag shared queries here), 'My Queries', and 'Shared Queries'. The main area is titled 'Followed work items' and 'Results'. It features a table with columns for ID, Work Item Type, Title, and State. Three work items are listed: a Bug titled 'Slow response on form' (Resolved), a User Story titled 'Cancel order form' (Active), and another User Story titled 'Welcome page' (Active).

ID	Work Item Type	Title	State
3	Bug	Slow response on form	Resolved
2	User Story	Cancel order form	Active
1	User Story	Welcome page	Active

From this view, you can view all items you're following across all projects. Also, you can perform similar actions supported with a query results view, such as:

- Refresh the view
- Add or remove visible columns
- Sort the order of specific columns
- Filter results by text or tags
- Set work item pane
- Enter full screen mode.

You can also view and manage work that you're following from your Project pages. To learn more, see [Work across projects](#).

Try this next

[Add, update, and follow a work item](#)

Related articles

- [Manage personal notifications](#)
- [View and update work items via the mobile work item form](#)

Q: Can I add someone else to follow a work item or PR?

A: You can't add another team member to follow a work item or pull request at this time. You can subscribe them to get notified based on select criteria, such as when a work item is created or modified, or a pull request is created. For details, see [Manage team notifications](#).

Get started as a Stakeholder

9/19/2019 • 18 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, manage build and release pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team. For a quick overview of the features available to Stakeholders, see the [Features and functions available to Stakeholders](#) later in this article.

NOTE

For public projects, Stakeholder access gives users greater access to features. To learn more, see [Default roles and access for public projects](#). For information about working with pipelines, see these articles: [Build your GitHub repository](#) and [Build OSS repositories](#).

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items, view and approve pipelines, and view dashboards. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team.

Stakeholders are users with free but limited access to Azure DevOps features and functions. With Stakeholder access, you can add and modify work items. You can check project status and provide direction, feedback, feature ideas, and business alignment to a team.

Stakeholder access is one of several supported access levels as described in [About access levels](#).

NOTE

Azure Boards supports several Agile methods such as Kanban and Scrum. Depending on what methods your team uses, you'll want to become familiar with other tools that Azure Boards supports. This article focuses on getting familiar with work items and the Kanban board. For additional information, see [Related articles](#) at the end of this article.

Use this tutorial to learn how to do the following tasks:

- Sign in to a project
- Understand which work item types are available to your project
- Open the Kanban board and add a work item
- Add details, tags, or comments to a work item
- View the product backlog
- Find work assigned to you, or query for other work items
- Understand what features are and aren't available to users with Stakeholder access

Connect to the web portal of a project

You must have been added to the Azure DevOps project and been granted Stakeholder or higher access level.

1. Choose the link provided in the email invitation you should have received. Or, open a browser window and enter the URL for the web portal.

`http://dev.azure.com/OrganizationName/ProjectName`

`http://ServerName:8080/tfs/DefaultCollection/ProjectName` For example, to connect to the server named *FabrikamPrime* and project named *Contoso*, enter
`http://FabrikamPrime:8080/tfs/DefaultCollection/Contoso`.

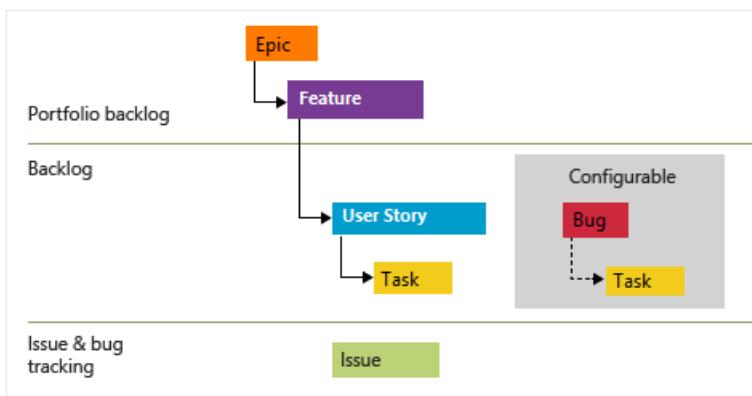
2. Enter your credentials. If you can't sign in, ask the organization owner or Project Administrator to add you as a member of the project with Stakeholder access.

Understand work items and work item types

Work items support planning and tracking work. Each work item represents an object stored in the work item data store. Each work item is based on a work item type and is assigned an identifier which is unique within an organization or project collection. Different work items are used to track different types of work as described in [About work items](#). The work item types available to you are based on the [process used when your project was created](#)—Agile, Basic, Scrum, or CMMI—as illustrated in the following images.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

User Stories and Tasks are used to track work, Bugs track code defects, and Epics and Features are used to group work under larger scenarios.



Open your Kanban board from the web portal

You can start viewing and adding work items once you connect to a project. The easiest way to add work items is through the backlog or Kanban board. Here we show how to add work through a Kanban board.

1. Check that you selected the right project, and select **Boards** > **Boards**. Then select the correct team from the team selector menu.

To select another team's board, open the selector. Then select a different team, or select the **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

TIP

Select the  star icon to make a team board a favorite. Favorite artifacts () appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level.



1. Check that you selected the right project, and select **Boards > Boards**. Then select the correct team from the team selector menu.



To select another team's board, open the selector. Then select a different team, or select the  **Browse all team boards** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.

**TIP**

Select the  star icon to make a team board a favorite. Favorite artifacts () appear at the top of the team selector list.

2. Check that you selected **Stories** for Agile, **Issues** for Basic, **Backlog items** for Scrum, or **Requirements** for CMMI as the backlog level. Here we have selected **Backlog Items** for the Scrum process.



1. To view your Kanban board, open your project from a web browser. Select **Work > Backlogs > Stories**, and then select **Board**.



If you don't see **Work**, your screen size might be reduced. Select the three dots  icon. Then select **Work > Backlogs > Board**.



2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.



Your Kanban board appears.



1. To view your Kanban board, open your project from a web browser. Select **Work > Backlogs > Stories**, and then select **Board**.



If you don't see **Work**, your screen size might be reduced. Select the three dots  icon. Then select **Work > Backlogs > Board**.

2. To select another team, open the project and team selector. Select a different team, or select the **Browse** option.



Your Kanban board appears.



Add work items to a product backlog

From the Kanban board, you can add work items. However, with Stakeholder access, you can't update the status of a work item by drag and drop to a different column.

- [Agile process](#)
- [Basic process](#)
- [Scrum process](#)
- [CMMI process](#)

1. From the User Stories board, choose **New item** and start adding those stories you want to track.



2. Enter return and the system assigns a work item ID to the user story.



3. Add as many user stories that you want to track.

Add details to a work item

To add information to a work item, open it by double-clicking the title or by selecting it and then typing Enter. Change one or more field values, add a description, [add a tag](#), or add a comment in the **Discussion** section. You can also choose the **Attachments** tab and drag-and-drop or upload a file to share with others.

To add information to a work item, open it by double-clicking the title or by selecting it and then typing Enter. Add a description, change one or more field values, or [add a tag](#). You can also choose the **Attachments** tab and upload a file to the work item to share with others.

You can only assign work to a user who has been added to the project.

NOTE

The work item form you see may differ from those shown in the following images. The basic functionality is the same, however, changes have been made with different versions of Azure DevOps.

- Agile process
- Basic process
- Scrum process
- CMMI process

For example, here we assign the story to Raisa Pokrovskaya and we add a discussion note, at-mentioning Raisa. Choose **Save & Close** when done.

The screenshot shows the 'USER STORY 1*' work item form. At the top, there's a header with a back arrow, a close button, and a title 'USER STORY 1*'. Below the header are buttons for 'Save & Close' (highlighted), 'Follow', and other options. The main area contains fields for 'State' (New), 'Reason' (New), 'Area' (Fabrikam Fiber), and 'Iteration' (Fabrikam Fiber). Below these are tabs for 'Details' and 'Related Work items'. The 'Description' section contains the text 'Switch initial view to the updated design.' The 'Acceptance Criteria' section has a placeholder 'Click to add Acceptance Criteria'. The 'Discussion' section shows a comment from 'Raisa Pokrovskaya' asking if it can be done in the next week, with a rich text editor below it. To the right, there are sections for 'Planning' (Story Points, Priority 2, Risk), 'Classification' (Value area: Business), 'Development' (dropdown), and 'Related Work' (dropdown).

Field descriptions

FIELD	DEFINITION
Title	Enter a description of 255 characters or less. You can always modify the title later.
Assigned To	Assign the work item to the team member responsible for performing the work. Depending on the context you are working in, the drop-down menu will list only team members or contributors to the project.
State	When the work item is created, the State defaults to the first state in the workflow. As work progresses, update it to reflect the current state.

Reason	Use the default first. Update it when you change state as need. Each State is associated with a default reason.
Area	Choose the area path associated with the product or team, or leave blank until assigned during a planning meeting. To change the dropdown list of areas, see Define area paths and assign to a team .
Iteration	Choose the sprint or iteration in which the work is to be completed, or leave it blank and assign it later during a planning meeting. To change the drop-down list of iterations, see Define iteration paths (aka sprints) and configure team iterations .
Description	Provide enough detail to create shared understanding of scope and support estimation efforts. Focus on the user, what they want to accomplish, and why. Don't describe how to develop the product. Do provide sufficient details so that your team can write tasks and test cases to implement the item.
Acceptance Criteria	Provide the criteria to be met before the user story can be closed. Before work begins, describe the customer acceptance criteria as clearly as possible. Conversations between the team and customers to define the acceptance criteria will help ensure that your team understands your customers' expectations. The acceptance criteria can be used as the basis for acceptance tests so that you can more effectively evaluate whether an item has been satisfactorily completed.
Priority	A subjective rating of the issue or task it relates to the business. You can specify the following values: 1: Product cannot ship without the successful resolution of the work item, and it should be addressed as soon as possible. 2: Product cannot ship without the successful resolution of the work item, but it does not need to be addressed immediately. 3: Resolution of the work item is optional based on resources, time, and risk. 4: Resolution of the work item is not required.
Value Area	The area of customer value addressed by the epic, feature, requirement, or backlog item. Values include: <ul style="list-style-type: none">• Architectural : Technical services to implement business features that deliver solution• Business: Services that fulfill customers or stakeholder needs that directly deliver customer value to support the business (Default)
Effort Story Points Size	Provide a relative estimate of the amount of work required to complete an issue. Most Agile methods recommend that you set estimates for backlog items based on relative size of work. Such methods include powers of 2 (1, 2, 4, 8) and the Fibonacci sequence (1, 2, 3, 5, 8, etc.). Use any numeric unit of measurement your team prefers. The estimates you set are used to calculate velocity and forecast sprints .

Add tags to a work item

Tags are useful for filtering backlogs, boards, and queries. As a Stakeholder, you can add existing tags to a work item, however, you can't add new tags.

From the web portal, open a work item and choose **Add tag** and type a keyword of an existing tag. Or, select from the list of previously assigned tags.

USER STORY 532

532 Hello World Web Site

Unassigned

0 comments

Add tag

State New

Area

Fabrikam Fiber A

Reason New

Iteration

Fabrikam Fiber A

Product Backlog Item 69: Hello World Web site



Tags page [Add...](#)

Hello World Web Site

Iteration Fabrikam Fiber Website

Area Fabrikam Fiber Website

Tags that appear in the tag bar are already assigned to the work item. To unassign a tag, simply choose the x on the tag, [Web](#).

NOTE

By default, all Contributors and Stakeholders of public projects are granted permissions to add new and existing tags.

Stakeholders in private projects can add tags that are already defined, but not add new tags. To grant or restrict permissions to create new tags, you set the permission **Create tag definition** at the project-level. To learn more, see [Add administrators, set permissions at the project-level or project collection-level](#).

Capture comments in the Discussion section

Use the **Discussion** section to add and review comments made about the work being performed.



The rich text editor tool bar displays below the text entry area when you click your cursor within each text box that can be formatted.



Mention someone, a group, work item, or pull request (, , or)

Choose one of these icons —, , or — to open a menu of recent entries you've made to mention someone, link to a work item, or link to a pull request. Or, you can simply type @, #, or ! to open the same menu.



NOTE

This latest version of the rich text editor requires Azure DevOps Server 2019 Update 1 or later version.

Type a name, or enter a number and the menu list will filter to match your entry. Choose the entry you want to add. You can bring a group into the discussion by typing @ and the group name, such as a team or security group.

Edit or delete a comment

If you need to edit or delete any of your discussion comments, choose **Edit** or choose the **⋮** actions icon and then choose **Delete**.

NOTE

The edit/delete feature requires Azure DevOps Server 2019 Update 1 or later version.

After updating the comment, choose **Update**. To delete the comment, you'll need to confirm that you want to delete it.

A full audit trail of all edited and deleted comments is maintained in the **History** tab on the work item form.

Use the **@mention control** to notify another team member about the discussion. Simply type @ and their name. To reference a work item, use the **#ID control**. Type # and a list of work items that you've recently referenced will appear from which you can select.

To reference a work item, use the **#ID control**. Type # and a list of work items that you've recently referenced will appear from which you can select.

IMPORTANT

For on-premises Azure DevOps Server or TFS, [you must configure an SMTP server](#) in order for team members to receive notifications.

Note that you can't edit or delete comments once they've been entered.

Add a reaction to a comment

You can add one or more reactions to any comment. Choose a smiley icon at the upper-right corner of any comment or choose from the icons at the bottom of a comment next to any existing reactions. To remove your reaction, click the reaction on the bottom of your comment. The following shows an example of the experience of adding a reaction, as well as the display of reactions on a comment.

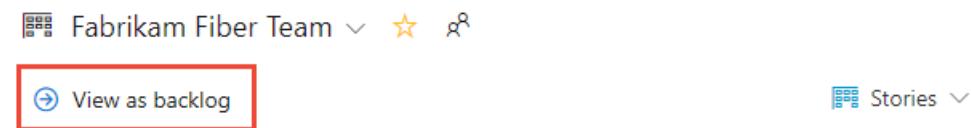
Check the backlog and prioritized work

You can check the product backlog to see how the team has prioritized work. Backlog items appear in priority order. Work item types may include bugs depending on the settings made for the team.

From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **View as backlog**.



From the Kanban board, choose **Backlog**.

The screenshot shows the 'Backlog items' page. At the top, there are tabs for 'Backlogs' and 'Queries'. Below them is a sidebar with 'Features' and 'Backlog items' (which is selected and highlighted with a red box). Under 'Current', it says 'Sprint 1'. The main area has a header with 'Backlog' (highlighted with a red box), 'Board', and various filtering options like 'Forecast Off', 'Mapping On', 'Parents Hide', 'In progress items Show', 'New', 'Create query', 'Column options', 'Filter', and a search icon. Below the header is a list of backlog items.

You should see the list of backlog items listed in priority order. You can add a backlog item which will be placed at the bottom of the list. With Stakeholder access, you can't re-prioritize work.

To view or edit a work item, select it and choose **Enter**.

Find work assigned to you, or query for other work items

1. Choose **Boards>Work Items**, and then select **Assigned to me**.

The screenshot shows the 'Work Items' page. At the top, there's a dropdown for 'Assigned to me' (highlighted with a red box) and buttons for 'New Work Item', 'Open in Queries', and 'Recycle Bin'. Below the dropdown is a section for 'Following', 'Mentioned', and 'My activity'. To the right, there are filters for 'Types' and 'States'. The main area lists work items with columns for 'with permissions', '...', 'State' (radio buttons for 'New' and 'In Progress'), and 'Assignee'. The first item is 'Architecture changes' assigned to 'Fabrikam Fiber\Voice'.

You can focus on relevant items inside a project using one of the seven pivots as described next.

Additionally, you can filter and sort each pivot view. For details, see [View and add work items using the Work Items page](#).

2. To query for work items, see [View, run, or email a work item query](#).

1. Open **Work>Queries** and select **Assigned to me** to see the list of work items assigned to you.

The screenshot shows the TFS 2015 interface for managing work items. The top navigation bar has 'HOME' and 'WORK' tabs, with 'WORK' selected. Below it, 'Backlogs' and 'Queries' are listed, with 'Queries' highlighted and circled in orange. A search bar at the top right says 'Search work items'. On the left, a sidebar contains sections for 'My favorites', 'Team favorites', 'My Queries', and 'Shared Queries'. Under 'Assigned to me', there are two work items listed:

ID	Work Item Type	Title	State
190	Bug	Simplify the search experience	New
191	Bug	Log-in button needs to be more prominent	New

Below the table, a specific bug is expanded, showing its details:

Bug 190: Simplify the search experience

Tags: Add...
Simplify the search experience

2. Or, open any of the queries defined in the Shared Queries folder.

The screenshot shows the TFS 2015 interface for managing work items. The top navigation bar has 'HOME' and 'WORK' tabs, with 'WORK' selected. Below it, 'Backlogs' and 'Queries' are listed, with 'Queries' highlighted and circled in orange. A search bar at the top right says 'Christie Church'. On the left, a sidebar contains sections for 'My favorites', 'Team favorites', 'My Queries', 'Shared Queries', and 'Current Sprint'. Under 'Current Sprint', 'Work in Progress' is selected and highlighted with an orange box. A query titled 'Work in progress' is displayed:

Work in progress

Results Editor

ID	Work Item Type	State	Remaining Work
164	Task	In Progress	8
165	Task	In Progress	8
166	Task	In Progress	6
167	Task	In Progress	2
168	Task	In Progress	2
169	Task	In Progress	1
170	Task	In Progress	4
173	Task	In Progress	2
174	Task	In Progress	1.5
181	Task	In Progress	1
186	Task	In Progress	1

3. And, you can [create new queries or edit existing queries](#) and save them under My Queries folder.

Features and functions available to Stakeholders

With Stakeholder access, users can create and modify work items and create and save queries. They have limited access to many of the Azure Boards features. They also can view and approve release pipelines and perform administrative tasks when granted administrative permissions or added to an administrative group.

NOTE

Stakeholders that choose a feature that's not available to them may in some instances receive an error message indicating that they don't have permissions to complete a task.

Public versus private feature access

Stakeholder access grants access to features differently depending on whether you're working from a private or a public project. To learn more about public projects, see [What is a public project?](#)

SERVICE, APPLICATION, OR SETTING	PRIVATE PROJECT	PUBLIC PROJECT
Dashboards	Partial access	Full access
Wiki	Partial access	Full access
Azure Boards	Partial access	Full access
Azure Repos	No access	Full access
Azure Pipelines	Full access	Full access
Azure Test Plans	No access	No access
Azure Artifacts	Full access	Full access
Notifications	Full access	Full access
Semantic search	Full access	Full access
Project settings	Partial access	Partial access

Service, application, or setting	Private project	Public project
Organization settings	Partial access	Partial access

Features not available to users with Stakeholder access

If a Stakeholder needs access to one or more of the following features—which support the daily work of product owners, team leads, developers, testers, and project administrators—you need to have **Basic** access.

For Private projects:

- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Pipelines, Repos or Test Plans**.

For Public projects:

- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Repos or Test Plans**.
- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify dashboards
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Pipelines, Repos, Test Plans, or Artifacts**.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify dashboards
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Code, Build and Release or Test**.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify dashboards
- View Delivery Plans (a Marketplace extension)
- Access the full set of features provided under **Code, Build and Release or Test**
- Participate in team rooms, which capture interactive, detailed conversations about the project.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify the home page
- Access the full set of features provided under **Code, Build and Release or Test**
- Participate in team rooms, which capture interactive, detailed conversations about the project.

Related articles

For a comparison chart of Stakeholder vs Basic access, see this [feature matrix](#). See also these quickstart guides:

- [Add work items](#)
- [Create your backlog](#)
- [Kanban quickstart](#)
- [Access levels](#)

- Change access levels

Change individual or group permissions

8/1/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

The standard way to set permissions is by adding them to one or more built-in security groups. However, sometimes you may want to grant additional permissions to select users, where not all permissions are assigned to the security group. For example, if you want to give some users the ability to add or edit area and iteration paths, but don't want them to have all permissions available to members of the Project Administrators group.

You can change individual permissions in one of the following three ways:

- Create a custom Azure DevOps security group, define permissions for that group, add the user account to the group
- For object-level permissions: Add the user account and set permissions
- For project or collection-level permissions: Search for the user account and selectively change their permission assignments

In this article you learn how to do the following tasks:

- Create a custom security group
- Set permissions for a custom security group
- Add members to a custom security group
- Change the permission assignments for an individual user

If you're new to managing permissions and groups, review [About permissions and groups](#) to learn about permission states and inheritance.

NOTE

The images you see from your web portal may differ from the images you see in this article. These differences result from updates made to Azure DevOps Services or your on-premises deployment. However, the basic functionality available to you remains the same unless explicitly mentioned.

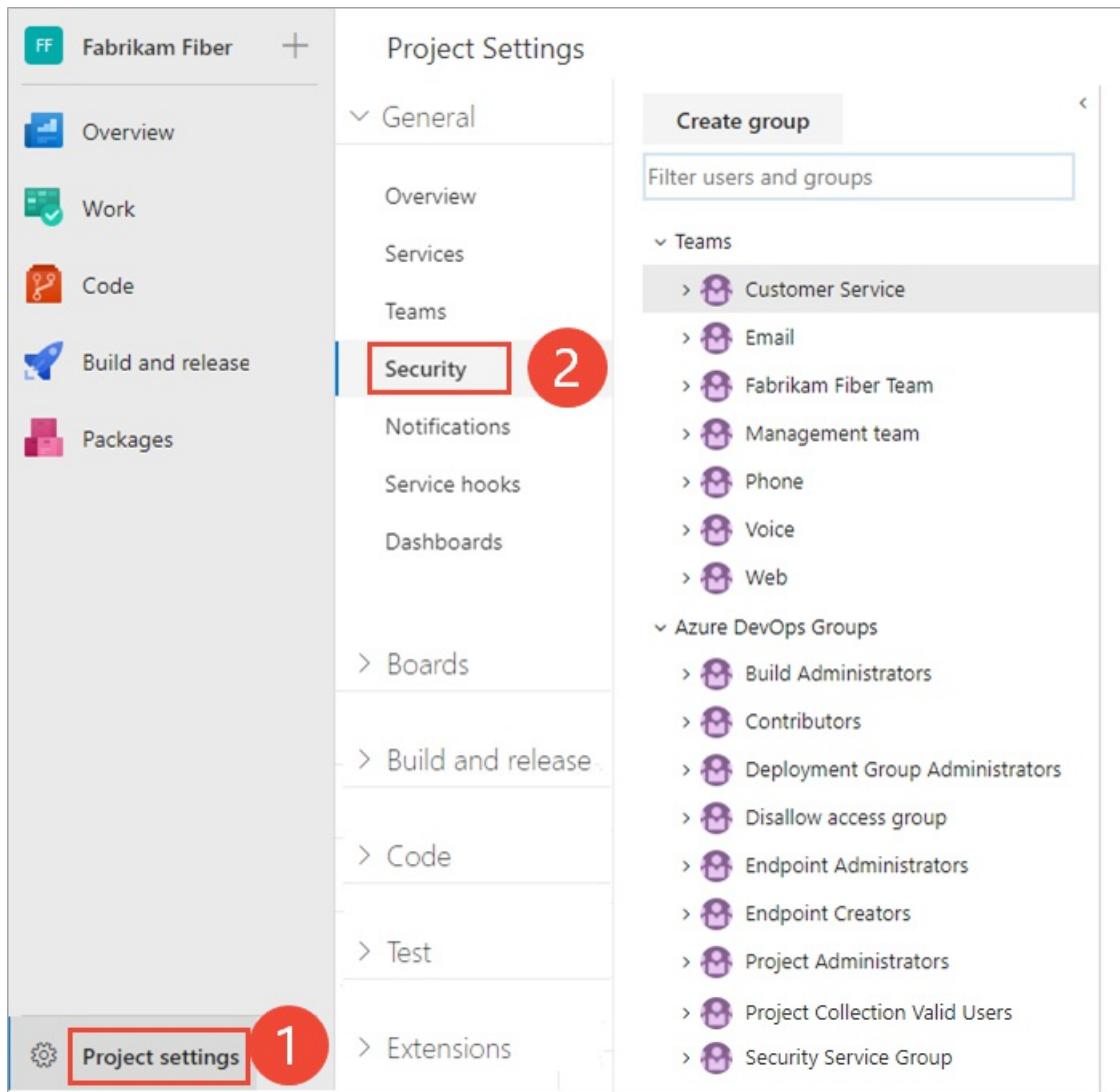
Create a custom security group

Create a custom security group at the project-level or the collection-level. The method for creating a custom security group is the same, no matter at what level you add it.

To create a project-level security group, open the web portal and choose the project where you want to add users or groups.

1. Choose **Project Settings > Security**.

To see the full image, click to expand.



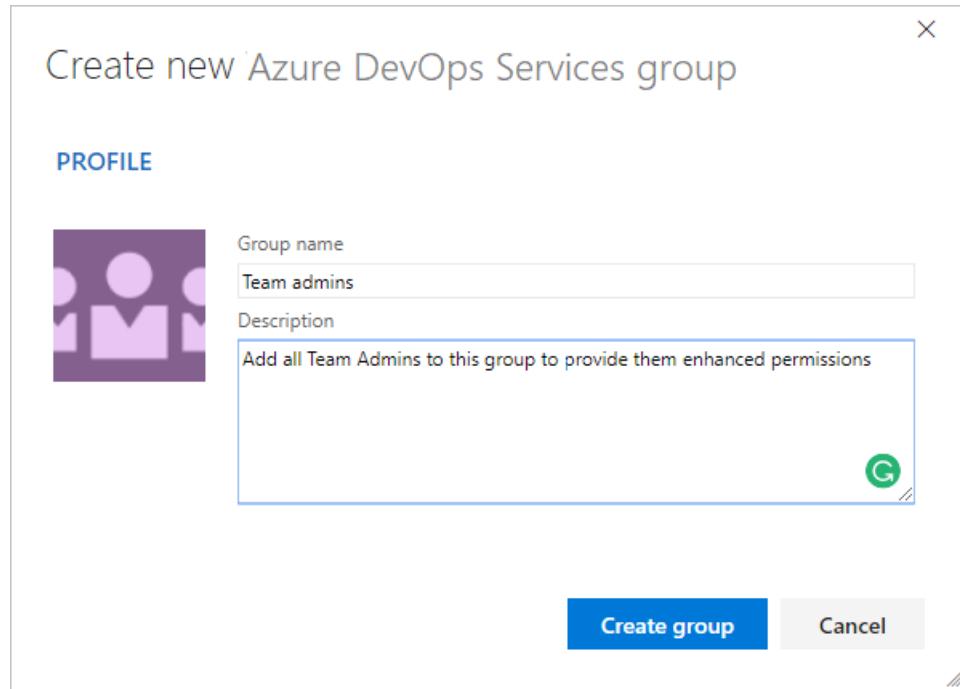
2. Choose **Create group** to open the dialog for adding a group.

The screenshot shows the 'Create group' dialog. At the top, there's a 'Create group' button and a 'Filter users and groups' input field. The 'Teams' section is expanded, showing 'Customer Service' selected. Other teams listed include Email, Fabrikam Fiber Team, Management team, Phone, Voice, and Web. Below the Teams section, the 'Azure DevOps Groups' section is expanded, listing various groups like Build Administrators, Contributors, Deployment Group Administrators, etc. On the right side of the dialog, there's a table showing permissions for the 'Customer Service' group. The table has columns for 'Permissions', 'Members', and 'Member of'. The permissions listed include: Bypass rules on work item updates (Not set), Change process of team project (Not set), Create tag definition (Allow (inherited)), Create test runs (Allow (inherited)), Delete and restore work items (Not set), Delete shared Analytics views (Allow (inherited)), Delete team project (Not set), Delete test runs (Allow (inherited)), Edit project-level information (Not set), Edit shared Analytics views (Allow (inherited)), Manage project properties (Not set), Manage test configurations (Allow (inherited)), Manage test environments (Allow (inherited)), Move work items out of this project (Not set), and Permanently delete work items (Not set).

Permissions	Members	Member of
Bypass rules on work item updates	Not set	
Change process of team project.	Not set	
Create tag definition	Allow (inherited)	
Create test runs	Allow (inherited)	
Delete and restore work items	Not set	
Delete shared Analytics views	Allow (inherited)	
Delete team project	Not set	
Delete test runs	Allow (inherited)	
Edit project-level information	Not set	
Edit shared Analytics views	Allow (inherited)	
Manage project properties	Not set	
Manage test configurations	Allow (inherited)	
Manage test environments	Allow (inherited)	
Move work items out of this project	Not set	
Permanently delete work items	Not set	

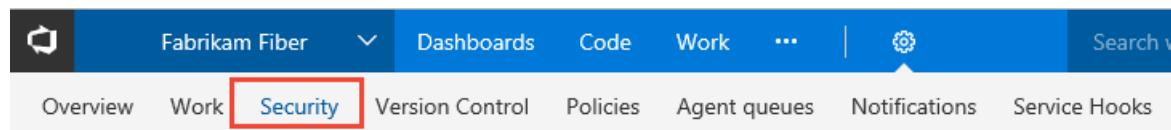
3. Enter a name for the group, and optionally a description.

For example, here we define a Team Admins group.



4. Choose **Create group**.

1. Open **Project Settings**. Choose the gear settings icon, and choose **Security**.



2. Choose **Create group** to open the dialog for adding a group.

The screenshot shows the "Create group" dialog. At the top left is a "Create group" button with a red box around it. Below it is a "Filter users and groups" input field. The main content area is divided into sections: "Teams" and "Azure DevOps Groups". Under "Teams", "Customer Service" is expanded, showing sub-items like "Email", "Fabrikam Fiber Team", etc. Under "Azure DevOps Groups", "Build Administrators", "Contributors", etc., are listed. To the right of these sections is a table of permissions for the "Customer Service" team:

Permissions	Members	Member of
Bypass rules on work item updates	Not set	
Change process of team project.	Not set	
Create tag definition	Allow (inherited)	
Create test runs	Allow (inherited)	
Delete and restore work items	Not set	
Delete shared Analytics views	Allow (inherited)	
Delete team project	Not set	
Delete test runs	Allow (inherited)	
Edit project-level information	Not set	
Edit shared Analytics views	Allow (inherited)	
Manage project properties	Not set	
Manage test configurations	Allow (inherited)	
Manage test environments	Allow (inherited)	
Move work items out of this project	Not set	
Permanently delete work items	Not set	

3. Enter a name for the group, and optionally a description.

For example, here we define a Team Admins group.

The screenshot shows the 'Create new Azure DevOps Services group' dialog. At the top left is a purple profile icon with three stylized human figures. To its right, the title 'Create new Azure DevOps Services group' is displayed. Below the title, the word 'PROFILE' is written in blue capital letters. On the left side of the main area, there is a large input field labeled 'Group name' containing the text 'Team admins'. To the right of this field is a smaller input field labeled 'Description' containing the text 'Add all Team Admins to this group to provide them enhanced permissions'. In the bottom right corner of the description field, there is a small green circular icon with a white letter 'G'. At the bottom of the dialog, there are two buttons: a blue 'Create group' button on the left and a grey 'Cancel' button on the right.

4. Choose **Create group**.

Set permissions for a custom security group

1. To set permissions for the custom group you created, choose the group name and then set one or more permissions.

The screenshot shows the 'Security' tab selected in the top navigation bar. On the left, there's a sidebar with a 'Create group' button and a 'Filter users and groups' input field. Below that is a tree view of security groups under 'Azure DevOps Groups': 'Build Administrators', 'Contributors', 'Disallow access group', 'Project Administrators', 'Project Valid Users', 'Readers', 'Release Administrators', and 'Team Admins'. The 'Team Admins' node is highlighted with a red box. The main panel shows the 'Fabrikam Fiber > Team Admins' group details. It has tabs for 'Permissions', 'Members', and 'Member of'. The 'Permissions' tab lists various project-level permissions with their current status: Allow, Deny, or Not set. Some permissions include 'Bypass rules on work item updates', 'Create tag definition', 'Delete and restore work items', 'Delete team project', etc. At the bottom of the permissions list is a 'Clear explicit permissions' link. At the very bottom are 'Save changes' and 'Undo changes' buttons.

Permission	Status
Bypass rules on work item updates	Allow
Create tag definition	Allow
Create test runs	Allow
Delete and restore work items	Allow
Delete team project	Deny
Delete test runs	Not set
Edit project-level information	Not set
Manage project properties	Deny
Manage test configurations	Allow
Manage test environments	Allow
Move work items out of this project	Allow
Permanently delete work items	Allow
Rename team project	Deny
Suppress notifications for work item updates	Not set
View analytics	Allow (inherited)
View project-level information	Allow
View test runs	Allow

For a description of each permission, see [Permissions and groups reference](#), [project-level permissions](#).

2. Choose **Save changes**.

Add members to a custom security group

You add members to a custom security group in the same way you add users to a built-in group.

1. Choose the security group, choose **Members**, and then choose **Add**.

The screenshot shows the 'Create group' interface. On the left, there's a sidebar with 'Teams' and 'Azure DevOps Groups' sections. Under 'Azure DevOps Groups', several options are listed: Build Administrators, Contributors, Disallow access group, Project Administrators, Project Valid Users, Readers, Release Administrators, and Team Admins. 'Team Admins' is highlighted with a grey background. The main area is titled 'Fabrikam Fiber > Team Admins'. It has tabs for 'Permissions' and 'Members', with 'Members' being the active tab. Below the tabs is a search bar with '+ Add...', 'Search', and refresh icons. The results table has columns for 'Display Name' and 'Username Or Scope'. A message says 'No identities found in current scope.'

2. Enter the user identity into the text box. You can enter several identities into the text box, separated by commas. The system automatically searches for matches. Choose the match(es) that meets your choice.

The screenshot shows the 'Add users and groups' dialog. It has a title 'Add users and groups' and a sub-instruction 'To add users or groups to this group, just type their sign-in addresses or group aliases'. A search input field contains 'Chris'. Below it, a search result card shows a profile picture of a woman, the name 'Christie Church', the email 'fabrikamfiber1@hotmail.com', and a copy icon. Below the card, it says 'Showing 1 result'. At the bottom are 'Save changes' and 'Cancel' buttons.

NOTE

Users that have limited access, such as Stakeholders, won't be able to access select features even if granted permissions to those features. To learn more, see [Permissions and access](#).

Change individual permission at the project-level

1. From the project-level **Security** page, enter the user identity in the **Filter users and groups** box. Then, select the account whose permissions you want to change.

The screenshot shows the 'Create group' interface again. The sidebar and main area are identical to the previous screenshot, but the search input field now contains 'Rais'. The search result card for Raisa Pokrovskaya is highlighted with a red box. The rest of the interface is the same, with tabs for 'Permissions' and 'Members', a search bar with '+ Add...', 'Search', and refresh icons, and a results table with 'Display Name' and 'Username Or Scope' columns.

2. Change the permission, setting a permission as **Allow** or **Deny**.

The screenshot shows the 'Create group' dialog for a user named 'Raisa Pokrovskaya'. The left sidebar lists 'Permissions' and 'Member of'. The main area displays a table of permissions with their current status. A blue box highlights the 'Allow' button for the 'Move work items out of this project' permission. At the bottom are 'Save changes' and 'Undo changes' buttons.

Permission	Status
Bypass rules on work item updates	Not set
Create tag definition	Allow (inherited)
Create test runs	Allow (inherited)
Delete and restore work items	Allow
Delete team project	Not set
Delete test runs	Allow (inherited)
Edit project-level information	Not set
Manage project properties	Allow
Manage test configurations	Allow (inherited)
Manage test environments	Allow (inherited)
Move work items out of this project	Allow
Permanently delete work items	Allow
Rename team project	Not set
Suppress notifications for work item updates	Not set
View analytics	Allow (inherited)
View project-level information	Allow (inherited)
View test runs	Allow (inherited)

For a description of each permission, see [Permissions and groups reference, project-level permissions](#).

3. Choose **Save changes**.

Change individual permission at the collection-level

1. Open the user-level or collection-level **Security** admin page and follow the instructions provided in the previous section for project-level permissions.

For a description of each collection-level permission, see [Permissions and groups reference, collection-level permissions](#).

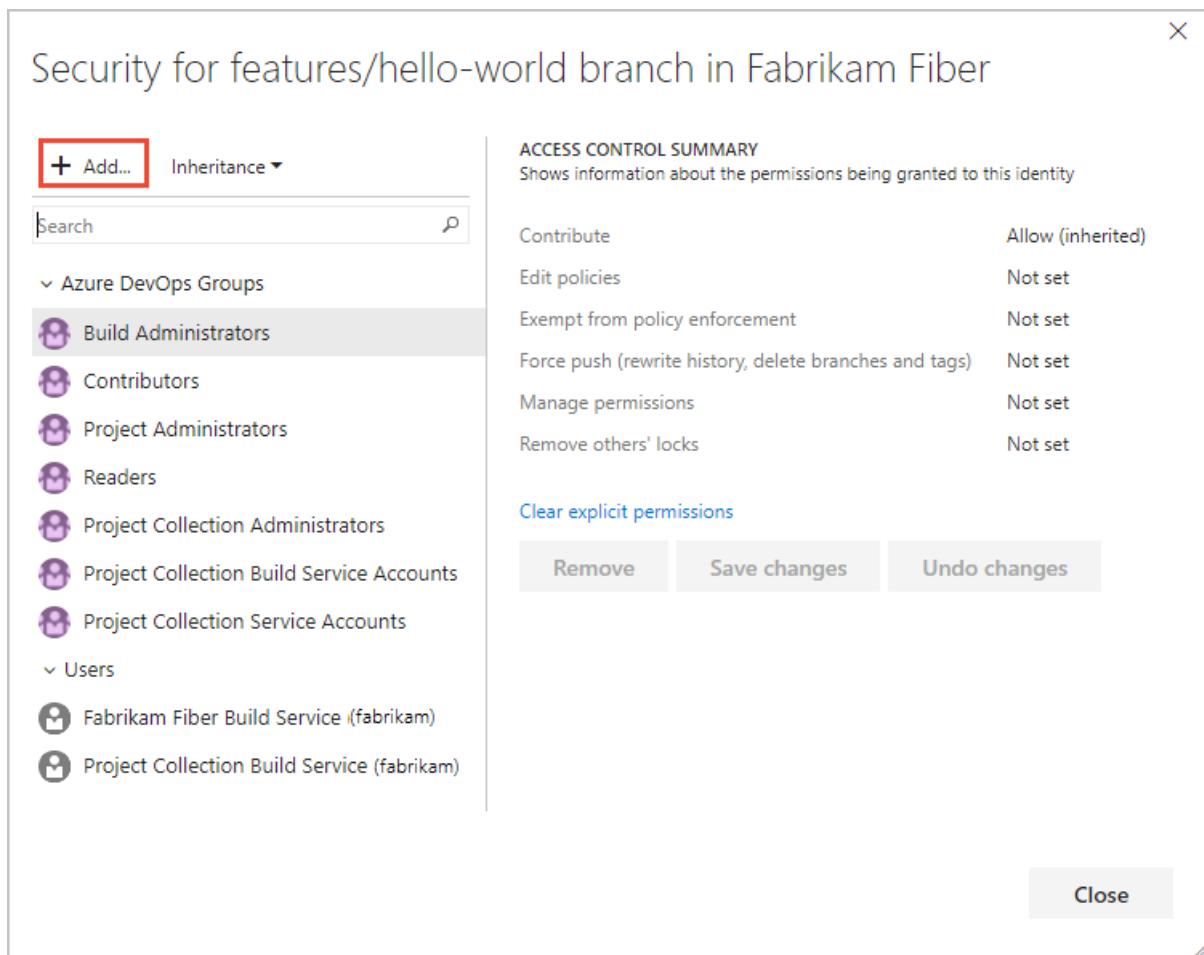
Change individual permission at an object-level

From the web portal, open the Security dialog for the object whose permissions you want to set. For specific instructions, see the following articles:

AREA	TASK
Wiki & Dashboard permissions	<ul style="list-style-type: none">• README & Wiki• Dashboards

DevOps (code, build, test, release) permissions	<ul style="list-style-type: none"> • Git branch • Git repository • TFVC • Builds • Release pipeline security • Approvals and approvers
Work tracking permissions	<ul style="list-style-type: none"> • Area and iteration paths • Work item query and folder • Plan permissions

1. From the Security dialog, choose **Add**.



2. Enter the user ID, choose search, and then make your selection in the left pane.
3. Update the permission setting to **Allow** or **Deny** for specific permissions.

Security for features/hello-world branch in Fabrikam Fiber

+ Add... Inheritance ▾

Search

✓ Azure DevOps Groups

- Build Administrators
- Contributors
- Project Administrators
- Readers
- Project Collection Administrators
- Project Collection Build Service Accou...
- Project Collection Service Accounts

✗ Users

- Raisa Pokrovskaya
- Fabrikam Fiber Build Service (fabrikam)
- Project Collection Build Service (fabrikam)

ACCESS CONTROL SUMMARY
Shows information about the permissions being granted to this identity

Contribute	Allow (inherited)
Edit policies	Allow (inherited)
Exempt from policy enforcement	Allow
Force push (rewrite history, delete branches and tags)	Allow (inherited)
Manage permissions	Allow (inherited)
Remove others' locks	Allow (inherited)

Clear explicit permissions

Remove Save changes Undo changes

Close

Action	Result
Contribute	Allow (inherited)
Edit policies	Allow (inherited)
Exempt from policy enforcement	Allow
Force push (rewrite history, delete branches and tags)	Allow (inherited)
Manage permissions	Allow (inherited)
Remove others' locks	Allow (inherited)

For a description of specific permissions, see [Permissions and groups reference](#).

4. Choose **Save changes**.

Next steps

[Grant or restrict access to select features](#)

Related articles

- [Permissions lookup guide](#)
- [About permissions and groups](#)
- [Permissions and groups reference](#)
- [Set permissions at the project-level or project collection-level](#)

Grant or restrict access

9/3/2019 • 8 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

You can grant or restrict access to resources that you manage in Azure DevOps. You may want to open up or close down access to a select set of features and for a select set of users. While the built-in security groups provide a standard set of permission assignments, you may need additional security requirements not met by these assignments.

If you're new to administering permissions and groups, review [About permissions and groups](#) to learn about permission states and inheritance.

In this article you learn how to do the following tasks:

- Recommended method for granting and restricting permissions
- Delegate tasks by assigning select permissions to specific roles
- Restrict access to view or modify objects
- Restrict modification of work items based on a user or group

TIP

Because you set many permissions at an object-level, such as repositories and area paths, how you structure your project determines the areas you can open up or close down.

Recommended method for granting and restricting permissions

For maintenance purposes, we recommend you use either the built-in security groups or [custom security groups to manage permissions](#).

You can't change the permission settings for the Project Administrators group or the Project Collection Administrators group, which is by design. However, for all other groups, you can change the permissions.

If you manage a small number of users, then you may find changing individual permissions a valid option. However, custom security groups allow you to better track roles and permissions assigned to those roles.

Delegate tasks to specific roles

As an administrator or account owner, it's a good idea to delegate administrative tasks to those team members who lead or manage an area. Several of the main built-in roles that come with default permissions and role assignments are:

- Readers
- Contributors
- Team Administrator (role)
- Project Administrators
- Project Collection Administrators

For a summary of permissions for the above roles, see [Default permissions and access](#), or for the Project Collection Administrators, see [Add administrators](#)

To delegate tasks to other members within your organization, consider creating a custom security group and then granting permissions as indicated in the following table.

Role	Tasks to perform	Permissions to set to Allow
Development lead (Git)	Manage branch policies	Edit policies, Force push, and Manage permissions See Set branch permissions .
Development lead (TFVC)	Manage repository and branches	Administer labels, Manage branch, and Manage permissions See Set repository permissions for Git or TFVC .
Software architect (Git)	Manage repositories	Create repositories, Force push, and Manage permissions See Set repository permissions for Git or TFVC .
Team administrators	Add area paths for their team Add shared queries for their team	Create child nodes, Delete this node, Edit this node See Create child nodes, modify work items under an area path Contribute, Delete, Manage permissions (for a query folder), See Set query permissions .
Contributors	Add shared queries under a query folder, Contribute to dashboards	Contribute, Delete (for a query folder), See Set query permissions View, Edit, and Manage dashboards, See Set dashboard permissions .
Project or product manager	Add area paths, iteration paths, and shared queries Delete and restore work items, Move work items out of this project, Permanently delete work items	Edit project-level information, See Add administrators, set permissions at the project-level or project collection-level .
Process template manager (Inheritance process model)	Work tracking customization	Administer process permissions, Create new projects, Create process, Delete field from account, Delete process, Delete project, Edit process See Add administrators, set permissions at the project-level or project collection-level .
Process template manager (Hosted XML process model)	Work tracking customization	Edit collection-level information, See Add administrators, set permissions at the project-level or project collection-level .
Project management (On-premises XML process model)	Work tracking customization	Edit project-level information, See Add administrators, set permissions at the project-level or project collection-level .

Permissions manager	Manage permissions for a project, account, or collection	<p>For a project, Edit project-level information For an account or collection, Edit instance-level (or collection-level) information To understand the scope of these permissions, see Permission lookup guide. To grant permissions, See Add administrators, set permissions at the project-level or project collection-level.</p> <p>You can also grant permissions to manage permissions for the following objects:</p> <ul style="list-style-type: none"> • Manage Git or TFVC repository permissions • Manage Git branch permissions • Administer build and release permissions • Manage Wiki permissions.
---------------------	--	---

Restrict access to view or modify objects

Azure DevOps is designed to enable all valid users to view all objects defined in the system. You can restrict access to resources by setting the permission state to **Deny**. You can set permissions for members that belong to a custom security group or for an individual user. To learn more about how to set these types of permissions, see [Change individual permissions, grant select access to specific functions](#).

Area to restrict	Permissions to set to Deny
View or contribute to a repository	View, Contribute See Set repository permissions for Git or TFVC .
View, create, or modify work items within an area path	Edit work items in this node, View work items in this node See Set permissions and access for work tracking , Modify work items under an area path .
View or update select build and release pipelines	Edit build pipeline, View build pipeline Edit release pipeline, View release pipeline You set these permissions at the object level. See Set build and release permissions .
Edit a dashboard	View dashboards See Set dashboard permissions .

Restrict modification of work items based on a user or group

Inheritance process model

For the [Inheritance process model](#), you can customize work item types to support these restriction requests:

- Restrict who can create or modify a work item
- Restrict who can create specific work item types, such as Epics or Features
- Restrict who can modify a specific field for a work item type

For example, the following condition indicates that the State field, for the Initiative custom work item type, becomes read-only for members of the Fabrikam Fiber\Voice group. When a user of this group opens a new Initiative, they are unable to save it as the State field can't automatically be set to New.

Conditions



The screenshot shows a 'When' condition step with the expression 'current user is member of group ...'. A dropdown arrow indicates more options. To the right is a preview box showing a user icon and the text '[Fabrikam Fiber]\V...'. A close button 'X' is also visible.

[+ New condition](#)

Actions



The screenshot shows a 'Then' action step with the expression 'Make read-only ...'. A dropdown arrow indicates more options. To the right is a preview box showing the text 'System.State'. A close button 'X' is also visible.

You can restrict modification of work items by adding a custom rule to the work item type. To learn more, see [Add a rule to a work item type \(Inheritance process\)](#).

Online XML process model

For the [On-premises XML process model](#), you can customize work item types to support these restriction requests:

- Restrict who can create or modify a work item
- Restrict who can create specific work item types, such as Epics or Features

For example, you can restrict modification of work items by adding a rule to the work item type, usually within the **WORKFLOW** section. To learn more, see [Add a rule to a work item type](#), [Apply or ignore rules based on user or group](#).

You restrict access to work tracking objects in one of two ways:

- Set a [condition field rule](#), a [condition-based field rule](#) or a combination of the two that applies to a group. You can restrict changes from being made to a field by specifying a qualifying rule and making it apply for a specific group. Conditional rules can include **CANNOTLOSEVALUE**, **EMPTY**, **FROZEN**, **NOTSAMEAS**, **READONLY**, and **REQUIRED** elements.
- By [adding WITs to the Hidden Categories group](#), you can prevent the majority of project contributors from creating them. You [can create a hyperlink to a template](#) that opens the work item form and share that link with those team members who you do want to create them.

Restrict modification of closed work items

Depending on your business processes, you may want to prevent users from continuing to modify or update work items that have been closed or completed. You can add rules to work item types to prevent users from re-opening closed work items.

To learn how, see this blog post, [Prevent reopening work item once closed](#).

NOTE

The blog post addresses how to add a custom rule to an Inherited process, however, the idea can be equally applied by adding one or more custom rules to an Online XML process and work item type definitions. To learn more, see [Apply a field rule](#).

Next steps

[Remove user accounts](#)

Related articles

- [Default permissions and access](#)
- [Permission lookup guide](#)
- [About permissions and groups](#)
- [Permissions and groups reference](#)
- [Set permissions at the project-level or project collection-level](#)

Plan your organizational structure

9/4/2019 • 15 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Your business structure should act as a guide to the number of organizations, projects, and teams that you create in Azure DevOps. This article helps you plan for different structures and scenarios for Azure DevOps.

Consider the following structures for your business or collaborative work in Azure DevOps:

- [Quantity of organizations](#)
- [Quantity of projects under an organization](#)

You also may want to plan for the following scenarios:

- [Mapping your organizations and projects](#) in Azure DevOps to your enterprise, business unit, and team structure
- [Structuring your repositories \(repos\)](#)
- [Structuring your teams](#)- it can either help or hinder teams to be Agile and autonomous
- [Managing access to data](#) - who needs to have access and who doesn't?
- [Reporting needs](#)
- Promoting common practices - learn more about [foundational elements you need to create an agile mindset and culture](#).

You need to have at least one organization, which may represent your company, your larger collection of code projects, or even multiple related business units.

What is an organization?

An organization in Azure DevOps is a mechanism for organizing and connecting groups of related projects. Examples are business divisions, regional divisions, or other enterprise structure. You can choose one organization for your entire company, or separate organizations for specific business units, or an organization just for you.

Each organization gets its own *free tier* of services (up to five users for each service type) as follows. You can use all the services, or choose just what you need to complement your existing workflows.

- [Azure Pipelines](#): One hosted job with 1,800 minutes per month for CI/CD and one self-hosted job
- [Azure Boards](#): Work item tracking and Kanban boards
- [Azure Repos](#): Unlimited private Git repos
- [Azure Artifacts](#): Package management
- Unlimited Stakeholders
 - Five Azure DevOps users (Basic)
 - Free tier of Microsoft-hosted CI/CD (one concurrent job, up to 30 hours per month)
 - 2GB of Azure Artifacts storage
 - One self-hosted CI/CD concurrent job
 - 20,000 virtual user minutes of cloud-based load testing

NOTE

The cloud-based load testing service is deprecated. More information about the deprecation, the service availability, and alternative services can be found [here](#).

How many organizations do you need?

When you're starting out with Azure DevOps, begin with one organization. Then, you can add additional organizations—which may require different security models—later. If you only have a single code repo or project, you don't need more than one organization. If you have separate teams that need to work on code or other projects in isolation, consider creating separate organizations for those teams. They'll have different URLs. Add projects, teams, and repos, as necessary, before you add another organization.

Take some time to review your work structure and the different business groups and participants to be managed. Read further for more guidance for [mapping your projects to business units](#) and [structure considerations](#).

What is a team?

A team is a unit that supports many [team-configurable tools](#), which help you plan and manage work, and make collaboration easier.

Creating a team for each distinct product or feature team

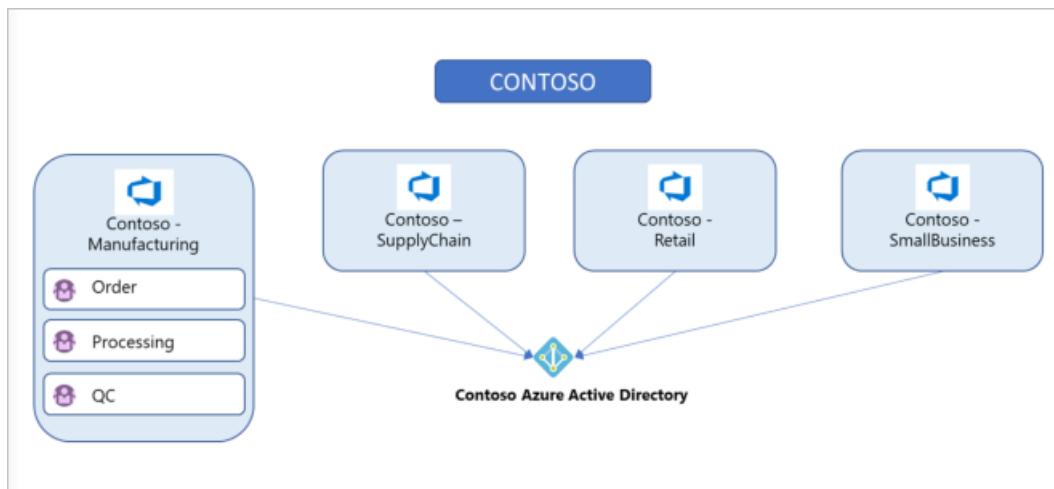
Every team owns their own backlog, to create a new backlog you create a new team. By [configuring teams and backlogs into a hierarchical structure](#), program owners can more easily track progress across teams, manage portfolios, and generate rollup data. A team group is created when you create a team. You can use this group in queries or to set permissions for your team.

What is a project?

A project in Azure DevOps contains the following set of features:

- Boards and backlogs for agile planning
 - Pipelines for continuous integration and deployment
 - Repos for version control and management of source code and artifacts
 - Continuous test integration throughout the project life cycle
- Each organization contains one or more projects

In the following image, the Contoso company has four projects within their Contoso-Manufacturing organization.



How many projects do you need?

You need at least one project to start using an Azure DevOps service, such as Azure Boards, Azure Repos, or Azure Pipelines. When you create your organization, a default project is created for you. In your default project, there's a code repo to start working in, backlog to track work, and at least one pipeline to begin automating build and release.

Within an organization, you can do either of the following approaches:

- Create a single project that contains many repos and teams
- Create many projects, each with its own set of teams, repos, builds, work items, and other elements

Even if you have many teams working on hundreds of different applications and software projects, you can manage them within a single project in Azure DevOps. However, if you want to manage more granular security between your software projects and their teams, consider using many projects. At the highest level of isolation is an organization, where each organization is connected to a single Azure AD tenant. A single Azure AD tenant can be connected to many Azure DevOps organizations.

Single project

A single project puts all of the work at the same "portfolio" level for the entire organization. Your work has the same set of repos and iteration paths. A single project allows teams to share source repos, build definitions, release definitions, reports, and package feeds. You might have a large product or service that's managed by many teams. Those teams have tight inter-dependencies on each other across the product life cycle. You create a project and divide the work using teams and area paths. This setup gives your teams visibility into each other's work, so the organization stays aligned. Your teams use the same taxonomy for work item tracking, making it easier to communicate and stay consistent.

TIP

When multiple teams work on the same product, having all teams on the same iteration schedule helps keep your teams aligned and delivering value on the same cadence. For example, the organization in Azure DevOps has over 40 feature teams and 500 users within a single project - this works well because we're all working on a common product set with common goals and a common release schedule.

A high volume of queries and boards can make it hard to find what you're looking for. Depending on the architecture of your product, this difficulty can bleed into other areas such as builds, releases, and repos. Make sure to use good naming conventions and a simple folder structure. When you add a repo to your project, consider your strategy and determine whether that repo could be placed into its own project.

Many projects

Project structure is best determined by how you ship the product. Having several projects shifts the administration burden and gives your teams more autonomy to manage the project as the team decides. It also provides greater control of security and access to assets across the different projects. Having team independence with many projects creates some alignment challenges, however. If each project is using a different process or iteration schedule, it can make communication and collaboration difficult if the taxonomies aren't the same.

TIP

If you use the same process and iteration schedules across all your projects, your ability to roll-up data and report across teams is improved.

Azure DevOps provides cross-project experiences when it comes to managing work.

You may want to add another project because of the following scenarios:

- To prohibit or manage access to the information within a project

- To support custom work tracking processes for specific business units within your organization
- To support entirely separate business units that have their own administrative policies and administrators
- To support testing customization activities or adding extensions before rolling out changes to the working project

When you're considering many projects, keep in mind that Git repo portability makes it easy to migrate repos (including full history) between projects. Other history can't be migrated between projects. Examples are push and pull request history.

When you map projects to business units, your company gets a single organization and sets up many projects with one or more projects representing a business unit. All Azure DevOps assets of the company are contained within this organization and located within a given region (for example, Western Europe). Consider the following guidance for mapping your projects to business units:

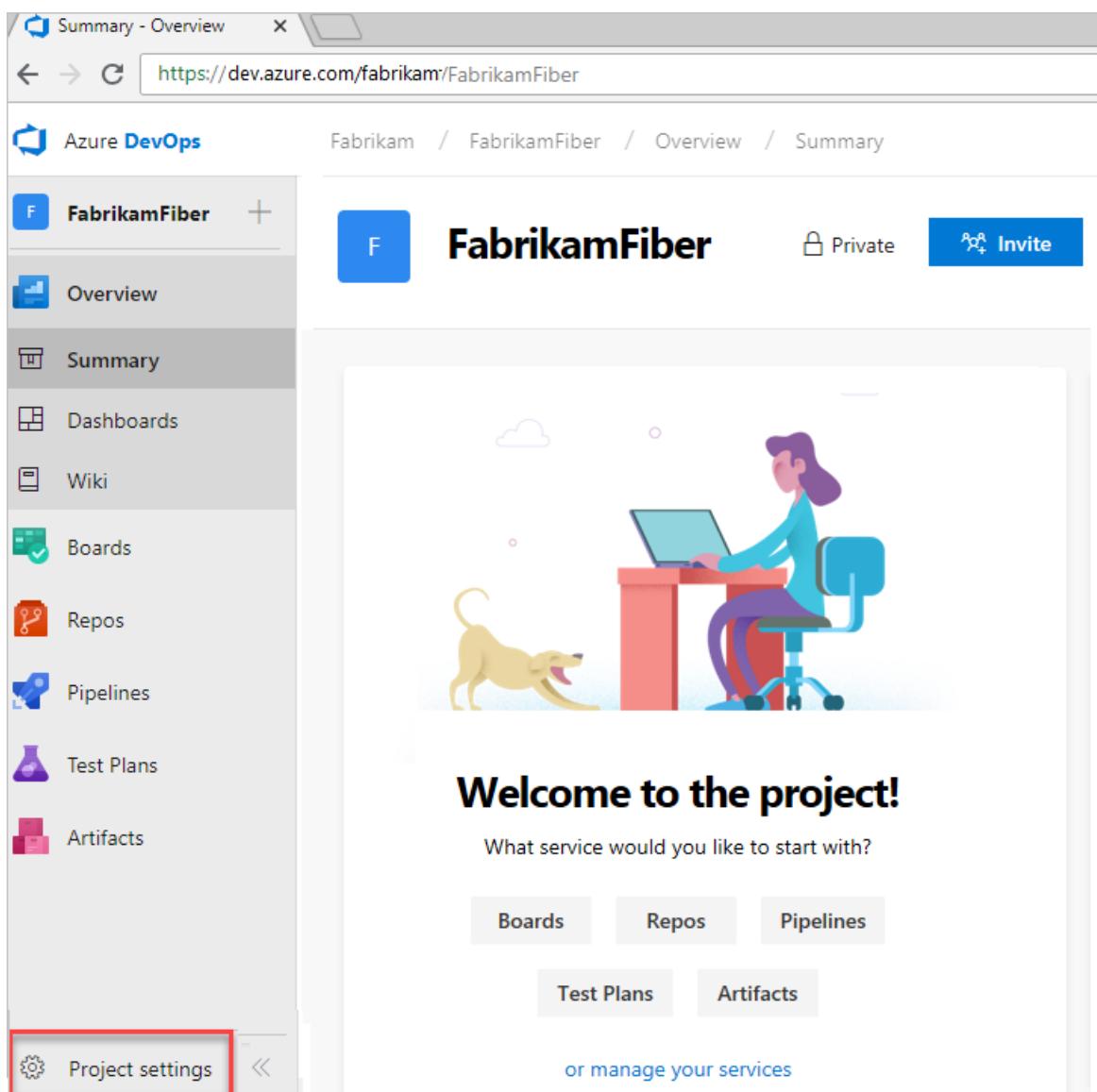
	ONE PROJECT, MANY TEAMS	ONE ORGANIZATION, MANY PROJECTS AND TEAMS	MANY ORGANIZATIONS
General guidance	Best for smaller organizations or larger organizations with highly aligned teams.	Good when different efforts require different processes.	Useful as part of TFS legacy migrations and for hard security boundaries between organizations. Used with multiple projects and teams within each organization.
Scale	Supports tens of thousands of users and hundreds of teams, but best at this scale if all teams are working on related efforts.	Same as with one project, but many projects may be easier.	
Process	Aligned processes across teams; team flexibility to customize boards, dashboards, and so on.	Independent processes for each project. For example, different work item types, custom fields, and so on.	Same as many projects.
Collaboration	Highest default visibility and reuse between work and assets of different teams.	Good visibility and reuse are possible, but it's easier to hide assets between projects whether intentional.	Poor visibility, collaboration, and reuse between organizations.
Roll-up reporting and portfolio management	Best ability to roll-up across teams and coordinate between teams.	Good reporting possible across projects. More difficult for cross-project roll-up and team coordination.	No roll-up or coordination between organizations.
Security/isolation	Can lock down assets at a team level, but default is open visibility and collaboration.	Better ability to lock down between projects. By default, provides good visibility within projects and good isolation across projects.	Hard boundaries across organizations; excellent isolation and minimal ability to share across organizations.
Context switching	Easiest for teams to work together and for users to switch between efforts.	Relatively easy for users to work together and switch contexts between efforts.	More difficult for users having to work across different organizations.

	ONE PROJECT, MANY TEAMS	ONE ORGANIZATION, MANY PROJECTS AND TEAMS	MANY ORGANIZATIONS
Information overload	By default, all assets are visible to users will make use of "favorites" and similar mechanisms to avoid "information overload."	Reduced risk of information overload; most project assets hidden across project boundaries.	Assets across organizations are isolated, reducing risk of information overload.
Administrative overhead	Much administration is delegated down to individual teams. Easiest for user licensing and org-level administration. Additional work may be needed if alignment is required between efforts.	Additional administration at the project level. Additional overhead, but can be useful when projects have different administrative needs.	As with additional projects, there's additional administrative overhead, which enables additional flexibility between orgs.

Structure repos and version control within a project

Consider the specific strategic work scoped to one of the organizations you created previously and who should have access. Use this information to name and [create a project](#). This project has a URL defined under the organization you created it in and can be accessed at <https://dev.azure.com/{organization-name}/{project-name}>.

Configure your project by visiting its URL and select the **Project settings** button at the lower right of the page.



To learn more about managing projects, see [Manage projects in Azure DevOps](#). You can move a project to a different organization by migrating the data. To learn more about migrating your project, see [Migration options](#).

Managing version control

In projects where the Azure Repos service is enabled, version control repos can store and revise code. Consider the following options when you're configuring repos.

Git vs. Team Foundation Version Control (TFVC)

Azure Repos offers the following version control systems for teams to choose from:

- Git and TFVC. Projects can have repos of each type. By default, new projects have an empty Git repo. Git enables a great amount of flexibility in developer workflows and integrates with nearly every relevant tool in the developer ecosystem. Any project can use Git repos. There's no limit on the number of Git repos that can be added to a project.

TFVC is a centralized version control system that is also available. Unlike Git, only one TFVC repository is allowed for a project. But, within that repo, folders, and branches are used to organize code for multiple products and services, if desired. Projects can use both TFVC and Git, if appropriate.

One vs. many repos

Do you need to set up multiple repos within a single project or have a repo set up per project? The following guidance relates to the planning and administration functions across those repos.

Starting with a single project containing multiple repos is reasonable, especially if the products/services are working on a coordinated release schedule. If developers are frequently working with multiple repos, keeping them in a single project encourages the processes to remain shared and consistent. Administering repo access is simpler in a single project, as access controls and options like case enforcement and max file size can be set at the project level. Repos can have these access controls and settings managed individually, even if they're in a single project.

If the products stored in multiple repos are working on independent schedules or processes, you can split them into multiple projects. Git repo portability makes it easy to move a repo between projects and still keep full-fidelity commit history. Other history, such as pull requests or build history, are not easily migrated.

Your decision for one vs. many repos should be largely based on code dependencies and architecture. A good first rule to apply is to put each independently deployable product or service in its own repo. Don't separate a codebase into many repos if you expect to make coordinated code changes across those repos, as there are no tools to assist in coordinating those changes. If your codebase is already a monolith, keep it in one repo. For more information about monolithic repos, see [Git at Scale](#) articles. If you have many disconnected services, one repo per service can be a good strategy.

NOTE

Consider [managing your permissions](#) so not everyone in your organization can [create a repo](#). One of the big challenges a growing team or company faces is the rapid proliferation of repos. If you have too many of them, it's very hard to keep track of who owns what code or other content stored in those repos.

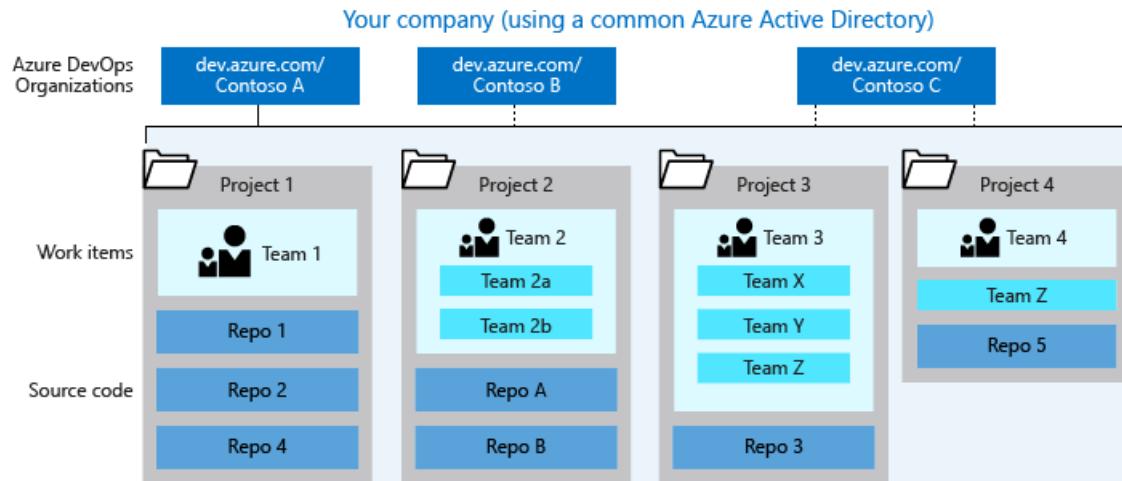
Shared repo vs. forked repos

We recommend using a shared repo within a trusted organization. Developers use branches to maintain isolation of their changes from one another. Used with a good branching and release strategy, a single repo can scale to support concurrent development for more than a thousand developers. For more information about branching and release strategy, see [Adopt a Git branching strategy and Release Flow: Our Branching Strategy](#).

Forks can be useful when you're working with vendor teams that shouldn't have direct access to update the main

repository. Forks can also be useful in scenarios where many developers contribute infrequently, such as in an open-source project. When you're working with forks, it may be useful to maintain a separate project to isolate the forked repos from the main repo. There may be added administrative overhead, but it keeps the main project cleaner. For more information, see the [Forks article](#).

The following image displays a sample of how "your company" could structure its organizations, projects, work items, teams, and repos.



More about organizational structure

Choosing your organization admin account type

When you create an organization, the identity that you sign in with defines the identity provider that the organization uses, such as your Azure Active Directory or Microsoft account. Your organization can be created by using a Microsoft account or with an Azure Active Directory account. This account provides the credentials to sign in as an admin to your new organization at <https://dev.azure.com/{yourorganization}>.

Using your Microsoft account

Use your Microsoft account if you don't need to authenticate users for an organization with Azure AD. All users must sign in to your organization with a Microsoft account. If you don't have a Microsoft account, you can [create a Microsoft account](#) at this time.

Microsoft

fabrikamfiber4@hotmail.com

Enter password

.....

Keep me signed in

[Forgot my password](#)

[Sign in with a different Microsoft account](#)

Sign in

If you don't have an Azure Active Directory instance, create one for free from the [Azure portal](#) or use your Microsoft account to create an organization. Then, you can [connect the organization to Azure AD](#).

Using your Azure Active Directory account

You might have an Azure AD account already if you use Azure or Office 365. If you work for a company that uses Azure AD to manage user permissions, you probably have an Azure AD account.

If you don't have an Azure AD account, learn how to [sign up for Azure AD](#) to automatically connect your organization to your Azure AD. All users must be members in that directory to access your organization. To add users from other organizations, use [Azure AD B2B collaboration](#).

Azure DevOps authenticates users through your Azure AD, so that only users who are members in that directory have access to your organization. When you remove users from that directory, they can no longer access your organization. Only specific [Azure AD administrators](#) manage users in your directory, so administrators control who accesses your organization.

After you create your Azure account, only members of that directory can access your organization, or you must use [Azure AD business-to-business \(B2B\) collaboration](#) to add users from other organizations.

Learn more about how to [manage users](#).

Mapping organizations to business units

Each business unit within your company gets its own organization in Azure DevOps, along with its own Azure Active Directory tenant. [Set up projects](#) within those individual organizations, as required, based on teams or ongoing work.

For a larger company, you can create multiple organizations using different user accounts (most likely Azure Active Directory accounts). Consider what groups and users share strategies and work, and group them into specific organizations. For example, the (fictional) Fabrikam company might create three organizations: Fabrikam-Marketing, Fabrikam-Engineering, and Fabrikam-Sales. Each organization has a separate URL, such as <https://dev.azure.com/Fabrikam-Marketing>, <https://dev.azure.com/Fabrikam-Engineering>, and <https://dev.azure.com/Fabrikam-Sales>. The organizations are all for the same company but are mostly isolated from each other. You don't need to have anything separated, however you should only create boundaries when it makes sense to your business. You can more easily partition an existing organization with projects, than combine different organizations.

Related articles

- [Create an organization](#)
- [Create a project](#)
- [Code with git](#)

What is source control?

9/26/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

A source control system, also called a *version control* system, allows developers to collaborate on code and track changes. Source control is an essential tool for multi-developer projects.

Our systems support two types of source control: Git (distributed) and Team Foundation Version Control (TFVC). TFVC is a centralized, client-server system. In both Git and TFVC, you can check in files and organize files in folders, branches, and repositories.

Manage your repos, branches, and other code development operations from **Azure Repos**.

Name	Last change	Date
.gitattributes	19 hours ago	11/18
.gitignore	11/18	11/18

With Git, each developer has a copy of the source repository on their dev machine. The source repo includes all branch and history information. Each developer works directly with their local repository. Changes are shared between repositories as a separate step.

Developers are able to commit each set of changes and perform version control operations, such as history and compare without a network connection. Branches are lightweight. When developers need to switch contexts, they create a private local branch. Developers can quickly switch from one branch to another to pivot among different variations of the code base. Later, developers can merge, publish, or dispose of the branch.

NOTE

Git in Visual Studio and Azure DevOps is standard Git. You can use Visual Studio with third-party Git services. You can also use third-party Git clients with TFS.

With TFVC, developers have only one version of each file on their dev machines. Historical data is maintained only on the server. Branches are path-based and are created on the server.

Next steps

Start sharing your code or get your code by using source control.

[Code with Git](#)

Related articles

- [Azure Repos documentation](#)
- [Git repositories documentation](#)

Tools and clients that connect to Azure DevOps

9/19/2019 • 9 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this article, learn about the tools and clients that connect to Azure DevOps.

Our platform of software development tools began more than 20 years ago. We released Visual Basic and Visual Studio as an integrated development environment (IDE). Visual Studio supports many plug-ins that extend its functionality. In particular, the Team Explorer plug-in allows the Visual Studio client to connect to Azure DevOps to support source control, work tracking, build, and test operations.

The available tools that communicate with Azure DevOps are included as follows:

- Desktop client developer tools
- Office integration tools
- Web-based tools
- Command-line tools
- Marketplace extensions
- REST APIs

Desktop client developer tools

Developers have access to many tools through these versions of Visual Studio and plug-ins. To download any version of Visual Studio, go to the [Visual Studio Downloads page](#). To understand what features you get with the Visual Studio versions, see [Compare Visual Studio offerings](#).

- **Visual Studio Community:** A fully featured and extensible IDE for creating modern applications for Android, iOS, and Windows, including web applications and cloud services. (Replaces Visual Studio Express.)
- **Visual Studio Professional:** Development tools and services to support individual developers or small teams.
- **Visual Studio Enterprise:** Integrated, end-to-end development tools and solutions for teams of any size, and with a need to scale. It supports designing, building, and managing complex enterprise applications.
- **Visual Studio Test Professional:** Provides access to Microsoft Test and development tools to support quality and collaboration throughout the development process.
- **Visual Studio Team Explorer:** Free solution for non-developers to interact with Team Foundation Server and Visual Studio Team Services.
- **Eclipse/Team Explorer Everywhere:** Free plug-in to support teams running Eclipse on Linux, macOS, or Windows that connects to Azure DevOps.
- **Android Studio with the Azure DevOps Services Plug-in for Android Studio:** Free plug-in to support Android developers and connect to Git repositories on Azure DevOps.
- **IntelliJ with the Azure DevOps Services Plugin for IntelliJ:** Free plug-in to support developers who use IntelliJ IDEA or Android Studio to connect to Git repositories on Azure DevOps.
- **Visual Studio Code:** Free, open-source code editor with a free extension to support connecting to Git repositories on Azure DevOps.

To get started with client libraries, see [Client library samples](#).

Team Explorer plug-in

Team Explorer, a plug-in to all Visual Studio versions, connects Visual Studio to projects defined in Azure DevOps.

You can manage source code, work items, and builds. To learn more, see [Work in Team Explorer](#).

HOME PAGE WITH GIT	HOME PAGE WITH TFVC

Office integration tools

You can integrate the following Microsoft Office tools with Azure DevOps.

- [Excel](#): Use Excel to add and bulk modify work items.
- [Project](#): By using Project, you can plan projects, schedule tasks, assign resources, and track changes. You have access to additional features, such as a project calendar, Gantt charts, and resource views.
- [Project Professional](#): With Project Professional, project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and easily share information.
- [Excel](#): Use Excel to add and bulk modify work items.
- [Project](#): By using Project, you can plan projects, schedule tasks, assign resources, and track changes. You have access to additional features, such as a project calendar, Gantt charts, and resource views.
- [Project Professional](#): With Project Professional, project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and easily share information.
- [PowerPoint Storyboarding](#): Illustrate user stories and requirements by using PowerPoint.

When you install any edition of Visual Studio or [Team Foundation Server Standalone Office Integration 2015 \(free\)](#), the Team Foundation plug-in integrates work item tracking with select Office clients. The Team Foundation plug-in installs to your existing Office client. The plug-in supports Office 2007, Office 2010, or Office 2013 versions.

- [Excel](#): Use Excel to add and bulk modify work items.
- [Project](#): By using Project, you can plan projects, schedule tasks, assign resources, and track changes. You have

access to features that TFS doesn't support, such as a project calendar, Gantt charts, and resource views.

- [PowerPoint Storyboarding](#): Illustrate user stories and requirements by using PowerPoint. The Team Foundation plug-in installs to your existing PowerPoint client.
- [Project Professional](#): With Project Professional and the Team Foundation Server Extensions for Project Server, you can manage projects that synchronize data that exists in both TFS and Project Server. Project managers and software development teams can use the tools that they prefer, work at the level of precision that supports their needs, and easily share information.

IMPORTANT

Support for integrating TFS with Project Server is deprecated for TFS 2017. However, synchronization support is provided by a Microsoft partner. See [Synchronize TFS with Project Server](#) for details.

Task-specific clients

The following clients support specific tasks, such as managing testing efforts, providing feedback, or modifying work items:

- [Azure Test Plans](#): Manage your test efforts, create and run manual tests, and create and track bugs that are found during test efforts. Test Plans is installed with Visual Studio Test Professional and Visual Studio Enterprise.
- [Test & Feedback extension \(previously called the Exploratory Testing extension\)](#): This extension provides a lightweight plug-in to a web browser. Stakeholders can respond to feedback requests for user stories and features created in Azure DevOps. This extension is free to Stakeholders.
- [Microsoft Feedback Client](#): Your Stakeholders can use this client to record feedback for your application as video, audio, or type-written comments. This client is installed with all versions of Visual Studio, or it can be [installed from the free download](#). All feedback is stored in the work item data store and requires [Stakeholders to have permissions](#).

IMPORTANT

Test Manager is deprecated for TFS 2017.

Browser-based web tools

Web portal

The collaboration tools supported through the web portal are summarized under [Essential services](#). New features are deployed every three weeks for Azure DevOps Services, and quarterly for Azure DevOps Server. For release notes, see [Azure DevOps Services Features Timeline](#).

You can use the following browsers to access the web portal:

VERSION	MICROSOFT EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
Azure DevOps Services	most recent	11 and later	9.1 and later	most recent	most recent
Azure DevOps Server 2019	most recent	11 and later	9.1 and later	most recent	most recent
TFS 2018	most recent	11 and later	9.1 and later	most recent	most recent

VERSION	MICROSOFT EDGE	INTERNET EXPLORER	SAFARI (MAC)	FIREFOX	CHROME
TFS 2015	most recent	9 and later	5 and later	most recent	most recent
TFS 2013		9 and later	5 and later	most recent	most recent

Microsoft Edge, Firefox, and Chrome automatically update themselves, so Azure DevOps supports the most recent version.

To learn more, see [Web portal navigation](#).

Browser-based extensions

The following extensions are available and are built and maintained by the Azure DevOps Services product team:

- [Azure Test Plans](#): Run tests by using your browser with simple pass/fail of steps, add comments/attachments, take screenshots, and file bugs. You can accomplish it all with automatic end-to-end traceability.
- [Azure Artifacts](#): Build packages of reusable code components and share them. The Azure Artifacts extension enables continuous delivery workflows by supporting multiple packaging protocols such as NuGet and npm. It makes packages available to your team, your builds, and your releases.
- [Code search](#): Increase cross-team collaboration and code sharing. Enables developers to quickly locate relevant information within the code base of all projects that are hosted within an organization or collection. You can discover implementation examples, browsing definitions, and error text.
- [Work item search](#): Quickly find relevant work items by searching across all work item fields over all projects in an organization. Do full-text searches across all fields to efficiently locate relevant work items. Use inline search filters, on any work item field, to quickly narrow down a list of work items.

Find additional extensions in Azure DevOps [Organization settings > Extensions > Browse marketplace](#).

Application monitoring tools

To monitor your applications, you can use Azure Application Insights for web apps or HockeyApp for mobile apps.

Monitor web applications with Application Insights

Application Insights is an extensible application performance management (APM) service for web developers. Use it to monitor your live web application. Application Insights automatically detect performance anomalies. It includes powerful analytics tools to help you diagnose issues and to understand what users actually do with your app. Application Insights is designed to help you continuously improve performance and usability. It works for apps on a wide variety of platforms—including .NET, Node.js, and Java EE—hosted on-premises or in the cloud.

With Application Insights, you can do the following tasks:

- Gain actionable insights through application performance management and instant analytics
- Detect and diagnose exceptions and application performance issues
- Monitor Azure websites, which includes websites hosted in containers, on-premises, and with other cloud providers
- Seamlessly integrate with your pipeline by using Azure DevOps Services, GitHub, and our webhooks
- Get started from within Visual Studio, or monitor existing apps without redeploying

To learn more, see [Microsoft Azure - Application Insights](#).

Monitor mobile applications with HockeyApp

With HockeyApp, you can develop, distribute, and beta test your mobile apps. HockeyApp supports the following apps and functionality:

- Android, Cordova, iOS, macOS, Unity, Windows, and Xamarin apps
- Live, reliable crash reports

- Collection of in-app feedback from real users
- Open-source SDKs to let you know what code is running in your apps
- Integration with your existing build system and work item management solution

To learn more, see [Microsoft Azure - HockeyApp](#).

Command-line tools

You can do many code development and administrative tasks by using the following command-line tools:

- [Git commands](#)
- [TFVC commands](#)
- [TFSConfig](#)
- [TFSDeleteProject](#)
- [TFS Security](#)
- [TFSServiceControl](#)
- [witadmin \(work item tracking\)](#)

Marketplace extensions

Visual Studio and Azure DevOps provide a wealth of features and functionality. They also provide a means to extend and share that functionality.

Extensions are simple add-ons that you can use to customize and extend your DevOps and work tracking experiences. They're written with standard technologies—HTML, JavaScript, and CSS. You can develop your own extensions by using your preferred dev tools.

You build extensions by using our RESTful API library. Publish your extensions to the Azure DevOps Marketplace. You can privately maintain or share them with millions of developers who use Visual Studio and Azure DevOps.

To learn more, visit the [Azure DevOps Marketplace](#) and see [Overview of extensions](#).

REST APIs

The Azure DevOps APIs are based on REST, OAuth, JSON, and service hooks—all standard web technologies broadly supported in the industry.

REST APIs are provided to support building extensions to Azure DevOps. To learn more, see [REST API overview](#).

Related articles

- [A tour of services](#)
- [Software development roles](#)
- [Pricing](#)
- [Azure DevOps data protection overview](#)

Software development roles supported by Azure DevOps

9/3/2019 • 4 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

If you're a sole developer or you work on a small team, you likely do tasks that are associated with issue tracking, feature planning, coding, testing, build, and deployment.

If you work in a large company, you're probably more focused on a specific set of tasks that are traditionally aligned with one or two specific roles, such as software development, project management, and DevOps.

This article describes the features and tasks available to you, based on your role.

Contributor roles

Team members are contributors who have access to the code base, work item tracking, Agile tools, build pipelines, test tools, and more. If you need to lock down specific areas to a select set of contributors, you can do that through [permission management](#).

Software developers

Developers use Visual Studio or other [tools](#) to develop their applications. They then check in their changes to a Git or Team Foundation Version Control (TFVC) repository hosted in Azure DevOps Services or Team Foundation Server (TFS). From the web portal or a supported IDE, they can view repositories, check history, and more.

- To get started with using Git, see one of the following resources:
 - [Share your code with Git and Visual Studio](#)
 - [Share your code in Git by using Eclipse](#)
 - [Share your code in Git by using Xcode](#)
 - [Share your code in Git by using IntelliJ](#)
 - [Get started with using Git and Azure DevOps Services](#)
- To get started with using TFVC, see one of the following resources:
 - [Develop and share your code in TFVC by using Visual Studio](#)
 - [Share your code in TFVC by using Eclipse](#)
 - [Share your code in TFVC by using Xcode](#)

Project managers

Project managers (PMs) typically plan the feature set to deliver, set priorities, and track the status of work, code defects, and customer issues. The suite of web-based Agile tools provides PMs with the views and features that they need to do these tasks. All work is captured within a work item. Each work item represents a specific type such as a user story, task, or bug.

- Use the product backlog to quickly define and prioritize user stories, features, and other work items
- Use the sprint backlog and task board to implement Scrum practices
- Use the Kanban board to work with Kanban methods
- Use queries to list and update work items, create status and trend charts, and post charts to dashboards
- Use dashboards to share information, status, and trends with your team or organization

For more information about getting started, see [About Azure Boards and Agile tools](#).

If you're used to using Excel or Project to plan and track your work, you can still use these tools and integrate with Azure DevOps. For more information, see [Bulk modify by using Excel](#) and [Create your backlog and tasks by using Project](#).

DevOps: builders, testers, and release managers

A main advantage of working with Azure DevOps is the suite of tools and integrated functionality that support build, testing, and deploying software applications. See the following main DevOps-associated tasks that Azure DevOps supports:

- Define builds
- Unit test your code
- Run tests with your builds
- Perform exploratory tests
- Define, manage, track, and approve releases
- Deploy applications to Azure, a virtual machine, Docker containers, and more

To get started, see the overviews in [Azure Pipelines](#) and [Azure Test Plans](#).

Stakeholders

With Stakeholder access, anyone in your organization can check project status and provide feedback. Stakeholders can track project priorities and provide direction, feature ideas, and business alignment to a team. Stakeholders also contribute to plans by adding and modifying work items. They can't, however, contribute to the code base or exercise test tools.

Stakeholder access essentially provides free access to a limited set of feature to project sponsors and supporters. To learn more, see [Work as a Stakeholder](#).

Administrator roles

A distinct advantage to working in Azure DevOps Services is the reduced overhead of server maintenance. But there are still several administrative tasks required to support a collaborative, integrated software development environment.

The main tasks are grouped as follows by membership in a security group or role.

Team administrators

Responsible for configuring team settings, which include:

- Backlog and board settings
- Team areas and iterations (sprints)
- Team members
- Team dashboards
- Team work item templates
- Team alerts

To get started, see [Manage teams and configure team tools](#).

Project administrators

Responsible for configuring project-level resources, including:

- [Area paths](#) and [iteration paths](#)
- [Project permissions and repository security](#)
- [Build agents, pools, and service connections](#)
- [Test and release retention policies](#)

- [Area paths](#) and [iteration paths](#)
- [Project permissions](#) and [repository security](#)
- [Customizing work tracking objects](#)
- [Build agents, pools, and service connections](#)
- [Test and release retention policies](#)

Organization Owners and Project Collection Administrators

Responsible for configuring organization-level resources, including the following tasks:

- Manage billing
- Add and manage projects
- Manage collection-level permissions
- Customize work tracking processes
- Install and manage extensions

To get started, see [Manage organizations](#) and [Settings](#).

Project Collection Administrators

Responsible for configuring collection-level resources. These tasks include:

- Add and manage projects
- Manage collection-level permissions
- Install and manage extensions

To get started, see [Settings](#).

TFS administrators

Responsible for installing, upgrading, and maintaining an on-premises TFS deployment, including the:

- Install TFS
- Update servers running TFS
- Manage database backups
- Manage server administrative settings and permissions
- Build retention policies
- Add and manage project collections

To get started, see [Server Administration \(TFS\)](#).

Related articles

- [A tour of services](#)
- [Plan your organizational structure in Azure DevOps](#)

Quickstart: Connect to a project in Azure DevOps

9/3/2019 • 7 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In this quickstart, you learn how to connect to a project in order to share code, build apps, track work, and collaborate with team members, from one of the following clients:

- [Web portal](#)
- [Visual Studio or Team Explorer](#)
- [Eclipse/Team Explorer Everywhere](#)
- [Android Studio with the Azure DevOps Services Plugin for Android Studio](#)
- [IntelliJ with the Azure DevOps Services Plugin for IntelliJ](#)
- [Visual Studio Code](#)

A project defines a process and data storage in which you manage your software projects from planning to deployment. When you connect to a project, you connect to an organization or project collection. Within that collection, one or more projects may be defined. At a minimum, at least one project must be created in order to use the system. For more information, see [About projects and scaling your organization](#).

Prerequisites

- If you don't have a project yet, [create one](#). If you need to add a team, see [Add teams](#). If you don't have access to the project, [get invited to the team](#).
- From each of these clients, you can quickly switch context to a different project and connect under a different sign-in user name. If you work remotely, configure your client to [connect to a TFS Proxy server](#).
- To get started with a code base, [set up Git](#) or [set up TFVC](#).

Connect from the web portal

1. If you're not a member of a security group, ask your Project Administrator to add you.
2. Open a browser window and enter a URL that uses the following form:

`https://dev.azure.com/OrganizationName/ProjectName`

`http://ServerName:8080/tfs/DefaultCollection/ProjectName`

For example, to connect to the server named **FabrikamPrime**, type: `http://FabrikamPrime:8080/tfs/`.

The default Port is 8080. Specify the port number and directory for your server if defaults aren't used.

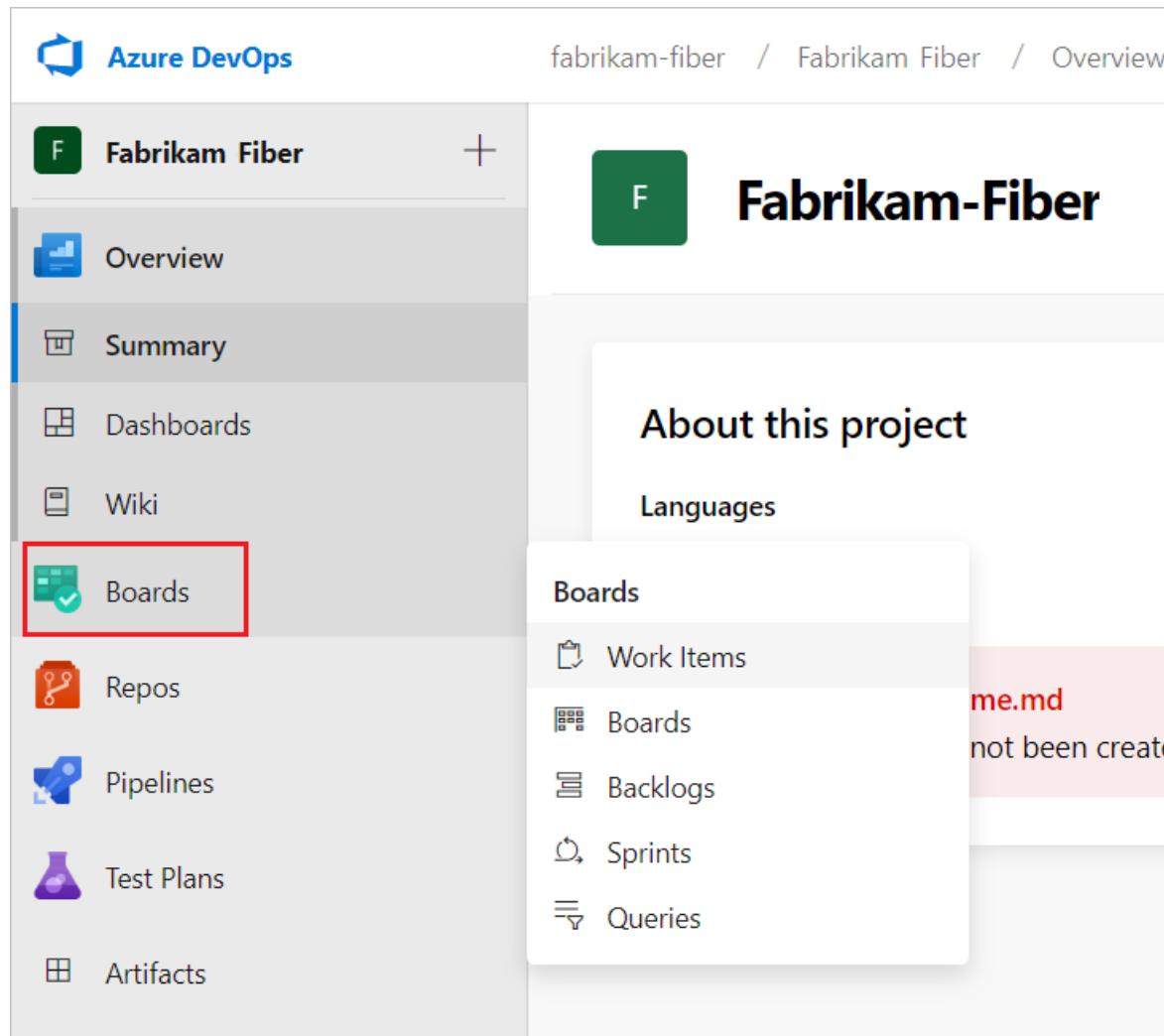
3. When you access the server for the first time, a Windows Identity dialog box appears. Fill in your credentials and choose the **OK** button.

TIP

If you select the **Remember me** check box you won't have to enter your credentials the next time you connect.

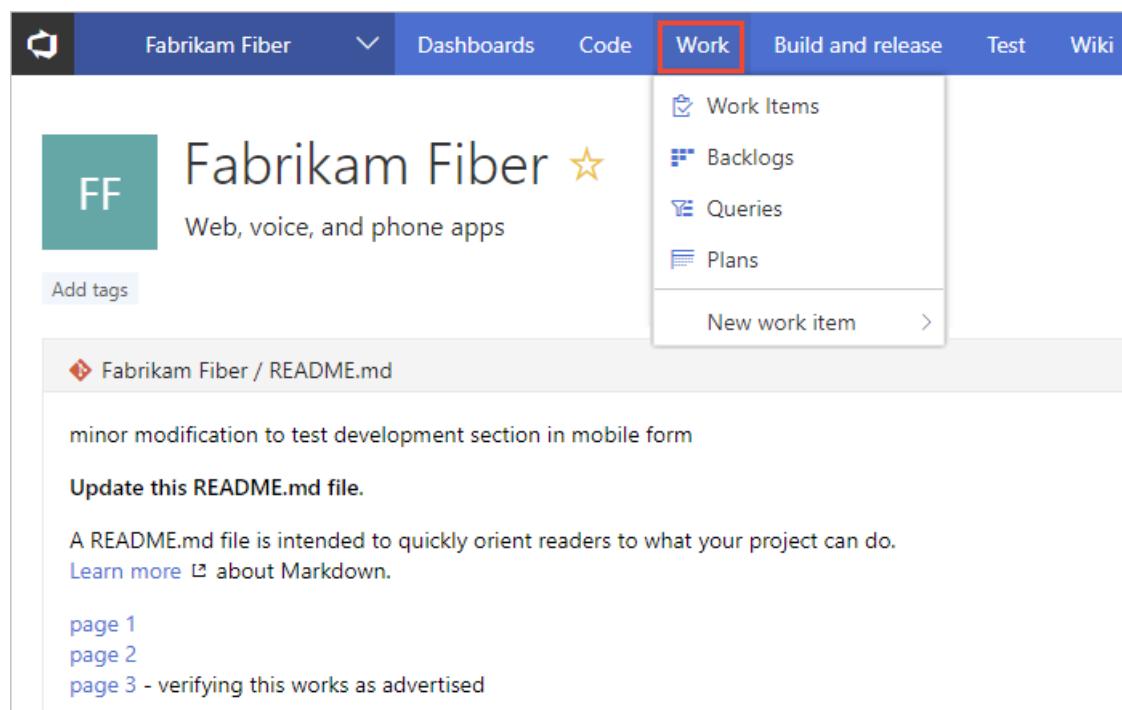
4. Choose your project, team, or page of interest.

From the project summary page, hover over a service and then choose the desired page. To choose another project, choose the  Azure DevOps logo.



The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, a sidebar lists various project services: Overview, Summary, Dashboards, Wiki, Boards, Repos, Pipelines, Test Plans, and Artifacts. The 'Boards' option is highlighted with a red box. In the main content area, the 'Fabrikam-Fiber' project is displayed. A tooltip is shown over the 'Boards' link in the main content area, listing 'Work Items', 'Boards', 'Backlogs', 'Sprints', and 'Queries'. The 'Boards' item is also highlighted with a red box.

From the project summary page, hover over a service and then choose the desired page. To choose another project, choose the  Azure DevOps logo.



The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. The top navigation bar includes tabs for 'Dashboards', 'Code', 'Work' (which is highlighted with a red box), 'Build and release', 'Test', and 'Wiki'. Below the navigation, the project summary for 'Fabrikam Fiber' is shown, featuring a teal square icon with 'FF', the project name, and a star icon. A tooltip is shown over the 'Work' tab, listing 'Work Items', 'Backlogs', 'Queries', and 'Plans'. The 'Work Items' item is also highlighted with a red box.

Choose your project or team from the set of available links, or choose Browse to access all projects and teams.

The screenshot shows the 'Overview' tab selected in the top navigation bar. Below it, the 'Rooms' tab is visible. The main content area is titled 'About Team Foundation Server' and contains four cards:

- Features**: What does Team Foundation Server have to offer? (with a 'View details' button)
- Learn**: Access online help for Team Foundation Server (with a 'View details' button)
- Get Visual Studio**: View all the download options (with a 'View details' button)
- Administer**: Manage projects, users, groups and permissions (with a 'View details' button)

Below these cards, there are two sections:

- Recent projects & teams**: A list of recent projects:
 - Fabrikam Fiber / Web Service (2 minutes ago)
 - Fabrikam Fiber (21 hours ago)
 - Fabrikam Fiber / Migrate (5/27/2016)
 - Fabrikam Fiber / Fiber Suite (2/3/2016)
- Recent team rooms**: A list of recent team rooms:
 - Fabrikam Fiber Team Room (0 users in room)

To learn more about each page and the tasks you can perform, see [Web portal navigation](#).

Sign in with different credentials

1. Open Windows Security from the context menu associated with your name.

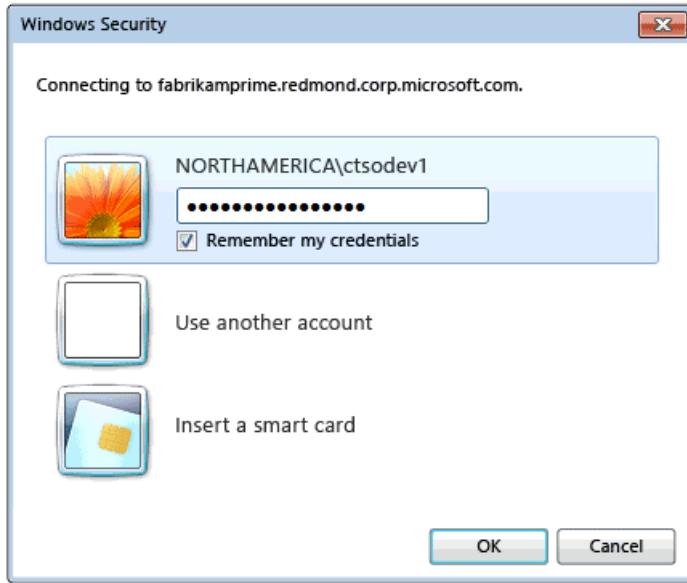
The screenshot shows the 'Overview' tab selected in the top navigation bar. The right side of the screen displays a user profile for 'Raisa Pokrovskaya'. A dropdown menu is open, showing the following options:

- My profile
- My alerts
- Sign in as...** (this option is highlighted with a red oval)
- Sign out

The user's name 'NORTHAMERICA\ctsodev1' is displayed at the bottom of the dropdown. The main content area on the left shows 'Visual Studio' links: 'Open in Visual Studio' (Requires Visual Studio 2013+) and 'Get Visual Studio' (See Visual Studio downloads).

2. Enter your credentials.

Sign In As...



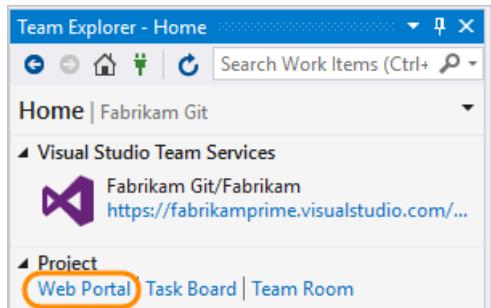
1. Open your profile menu and choose **Sign out**.

A screenshot of the Microsoft Teams profile menu. The user 'Jamal Hartnett' is logged in, with the email 'fabrikamfiber4@hotmail.com'. The menu includes links for 'My profile', 'Security', 'Usage', 'Notification settings', 'Preview features', 'Theme', and 'Help'. The 'Sign out' button at the bottom is highlighted with a red rectangle.

2. Choose Sign in and enter the new credentials.

Open the web portal from Team Explorer

- Open the web portal from the home page.

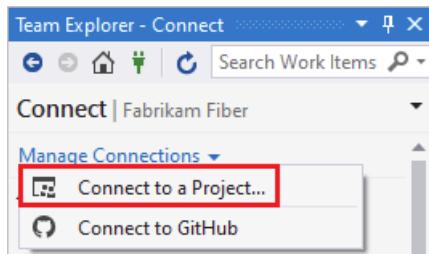


Connect from Visual Studio or Team Explorer

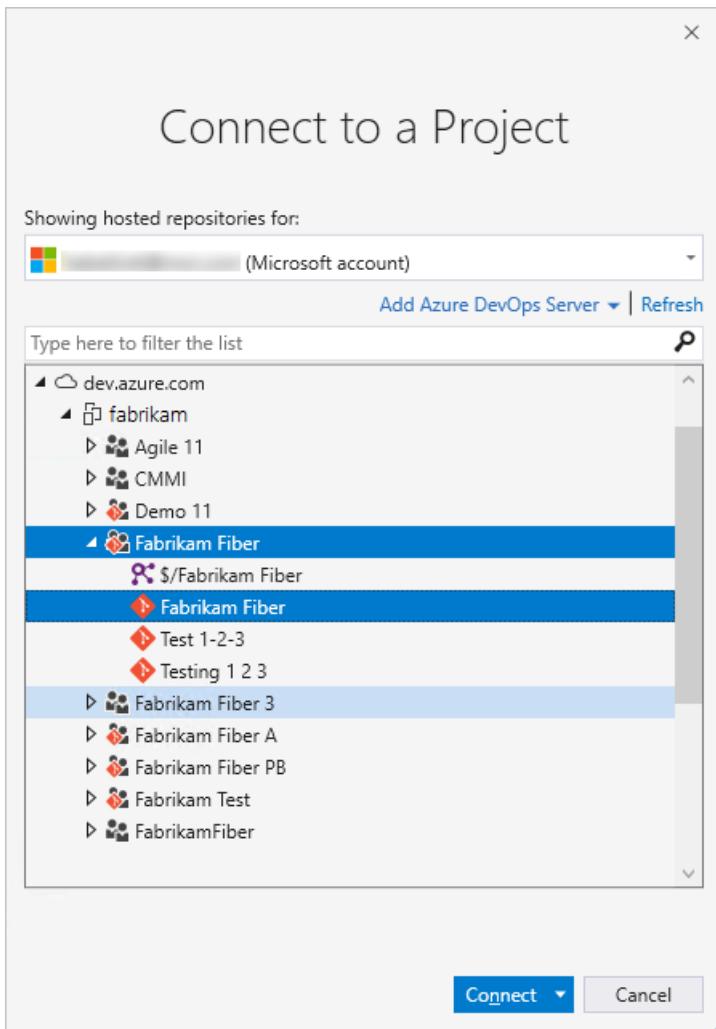
1. If you haven't already, [download and install a version of Visual Studio](#).
2. If you're not a member of an Azure DevOps security group, [get added to one](#).
3. Check with a team member to determine the names of the server, project collection, and project to connect to.
 - [Visual Studio 2019](#)
 - [Visual Studio 2017](#)
 - [Visual Studio 2015](#)

Visual Studio 2019

Select the connect icon in Team Explorer to open up the **Connect** page. Choose the **Connect to Team Project** link to select a project to connect to.



The **Connect to a Project** dialog appears and shows the projects you can connect to, along with the repos in those projects.



Select the **Add Azure DevOps Server** link to connect to a project in Azure DevOps Services. Enter the URL to your server and select **Add**.



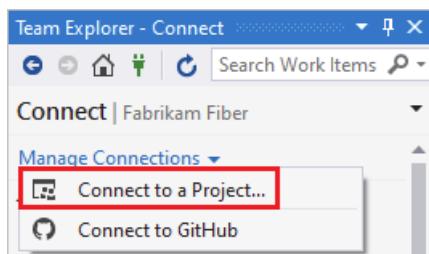
Select a project from the list and select **Connect**.

Change sign-in credentials

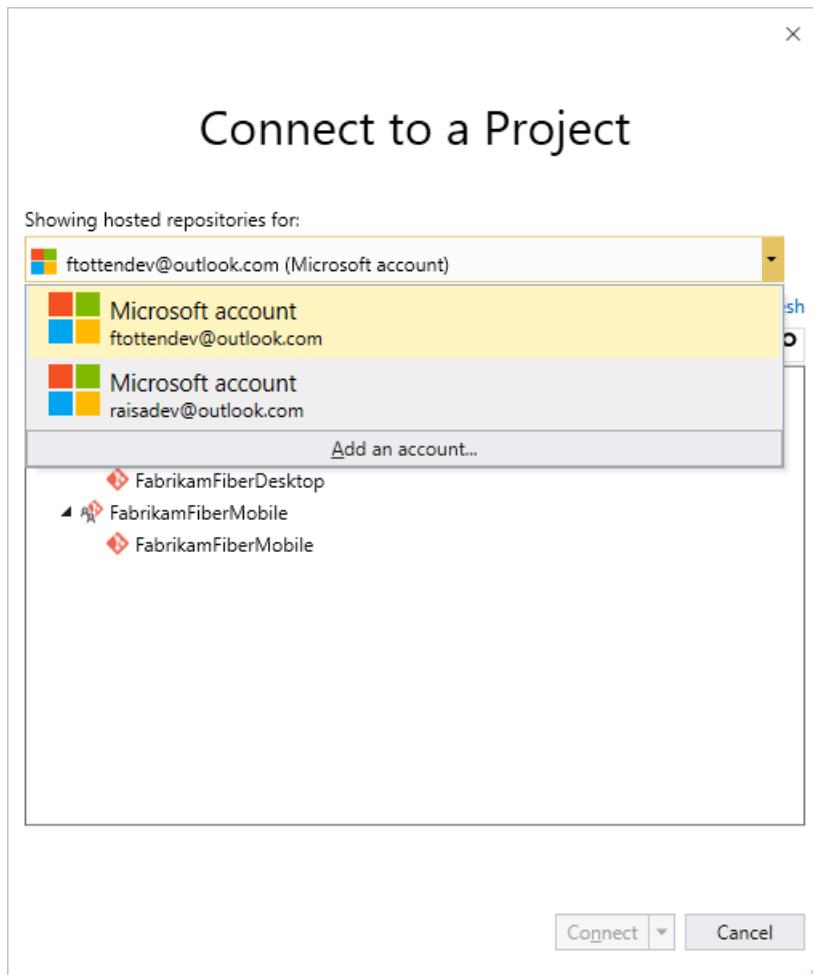
- [Visual Studio 2019](#)
- [Visual Studio 2017](#)
- [Visual Studio 2015](#)

Visual Studio 2019

1. From the Connect page, choose the **Connect to a Project** link to sign in with different credentials.

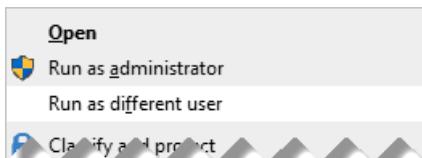


Select a different user from the drop-down or select **Add an account...** to access a project using different sign-in credentials.



- Sign in using an account that is associated with an Azure DevOps project, either a valid Microsoft account or GitHub account.

To run Visual Studio under sign-in credentials that are different from your signed-in Windows account, open the context menu for **devenv.exe** to access your run as options. If you don't see the **run as** option as shown in the following example, you may need to press SHIFT before right-clicking to see the run as options.



You can locate the executable in the following folder:

```
*Drive*:\\Program Files (x86)\\Microsoft Visual Studio xx.0\\Common7\\IDE\\
```

User accounts and licensing for Visual Studio

To connect to a project, you need your user account added to the project. This is typically done by the [organization owner \(Azure DevOps Services\)](#) or a [Project Administrator](#).

Azure DevOps Services provides access to the first 5 account users free. After that, you need to [pay for more users](#).

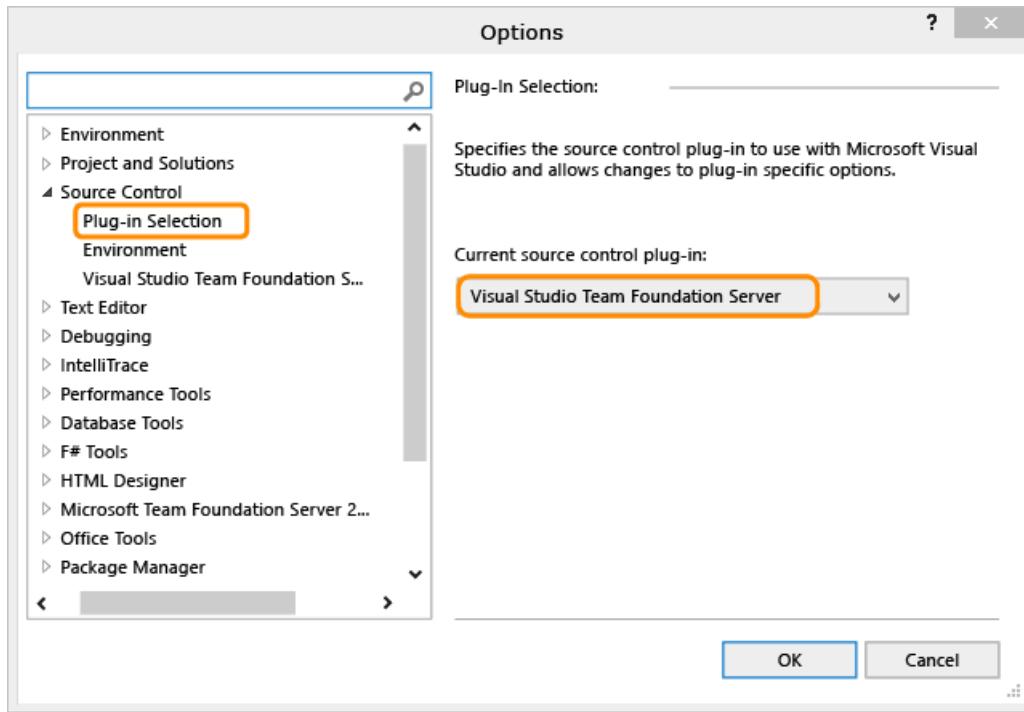
For on-premises TFS, each user account must have a TFS client access license (CAL). All Visual Studio subscriptions and paid Azure DevOps Services users include a TFS CAL. Find out more about licensing from the [Team Foundation Server pricing page](#).

In addition, you can provide access to Stakeholders in your organization who have limited access to select features as described in [Work as a Stakeholder](#).

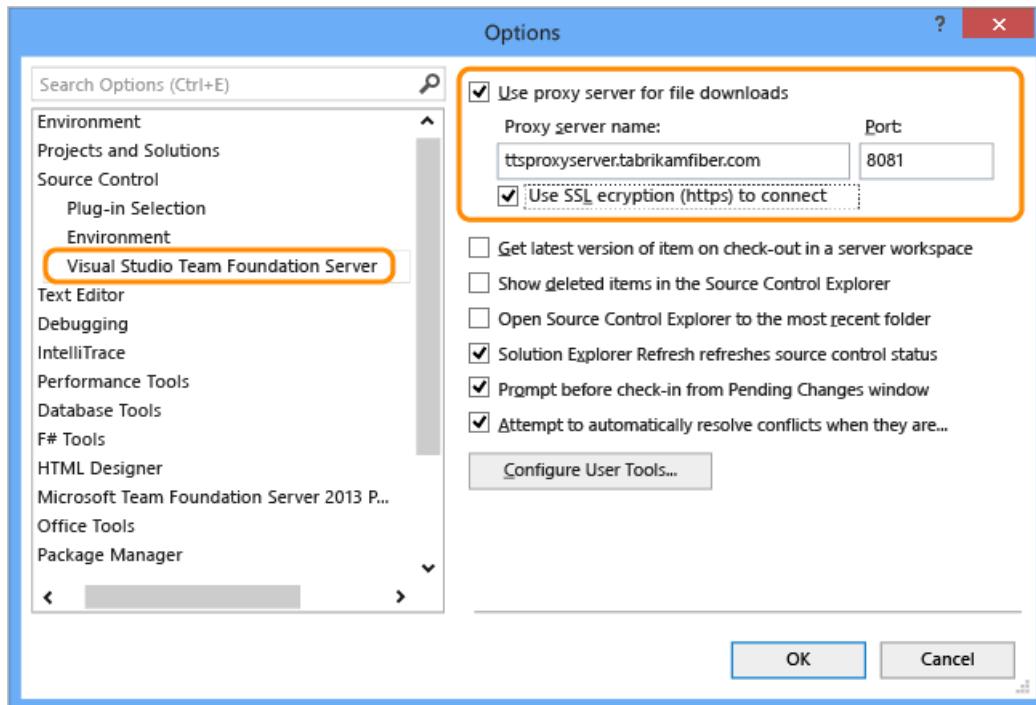
Configure Visual Studio to connect to TFS Proxy

If your remote team uses a [TFS Proxy server](#) to cache files, you can configure Visual Studio to connect through that proxy server and download files under Team Foundation version control.

1. First, make sure that you have connected to TFS as described [in the previous section](#).
2. From the Visual Studio **Tools** menu, open the Options dialog and expand the Source Control folder. On the Plug-in Selection page, confirm that Visual Studio Team Foundation Server is selected.



3. On the Visual Studio Team Foundation Server page, enter the name and port number for the TFS Proxy server. Select the **Use SSL encryption (https) to connect** checkbox.



Make sure you specify the port number that your administrator assigned to TFS Proxy.

To **Configure User Tools** to associate a file type with a compare or merge tool, see [Associate a file type with a file-comparison tool](#) or [Associate a file type with a merge tool](#).

What other clients support connection to Azure DevOps?

In addition to connecting through a web browser, Visual Studio, Eclipse, Excel, and Project you can connect to a project from these clients:

- [Visual Studio Code](#)
- [Visual Studio Community](#)
- [Eclipse: Team Explorer Everywhere](#)
- [Azure Test Plans](#) (formerly Test Manager)
- [Microsoft Feedback Client](#)

Requirements and client compatibility

Some tasks or features aren't available when you connect to a later version of Azure DevOps Server than which your client supports. For more information, see [Client compatibility](#).

Determine your platform version

See [Feedback and support](#).

Next steps

Learn more about how to:

- [Work in web portal](#)
- [Work in Team Explorer](#)
- [Work in Office Excel or Project](#)
- [Troubleshoot connection](#)

If all you need is a code repository and bug tracking solution, then start with the [Git get started guide](#) and [Manage bugs](#).

To start planning and tracking work, see [Get started with Agile tools to plan and track work](#).

Troubleshoot connecting to a project

8/1/2019 • 4 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

Troubleshoot connectivity

As a first step in resolving connectivity issues with Azure DevOps, complete the following steps:

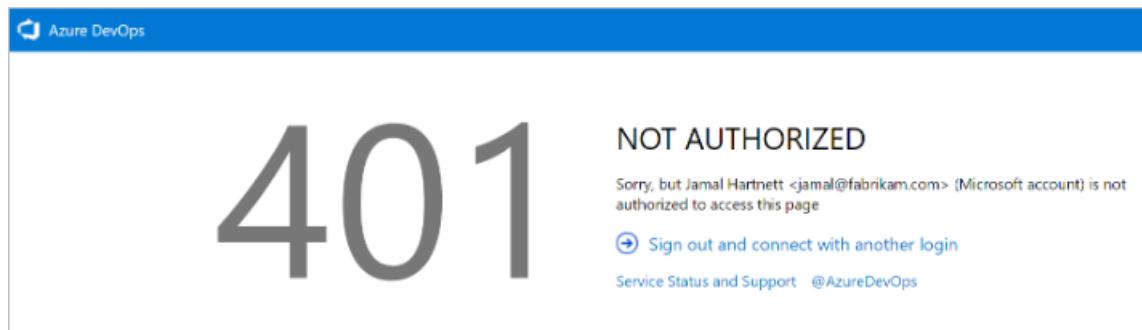
1. Sign out of your browser. To do so, select the [Visual Studio sign out](#) link.
2. Delete the cookies in your browser. To delete cookies in most browsers, press Ctrl+Shift+Del.
3. Open Internet Explorer and delete the browser cookies. The Visual Studio IDE uses Internet Explorer cookies.
4. Close all browsers and close the Visual Studio IDE.
5. Use a private browser session to retry the connection. If the issue is with the Visual Studio IDE, remove the connection, and then readd it.

Troubleshoot signing in

Two types of identities can sign in: Microsoft accounts and Azure Active Directory (Azure AD) accounts.

Depending on your account, you might experience one of the following errors.

401 - Not Authorized



The most common error page is the *401 Not Authorized* error, which occurs when your identity doesn't have permissions to enter an organization. Common reasons for the error include:

- Your identity isn't a member of the organization.
- Your identity has an invalid or missing license assignment.

If you think you're a member of the organization but are blocked by this error page, [contact customer support](#).

Scenario 1

Your work or school Azure AD account doesn't have access, but your personal Microsoft account does.

401 - Work or school, or Personal account

401

NOT AUTHORIZED

jamal@fabrikam.com has multiple accounts associated with it.

Your work or school account does not have access to dev.azure.com/Fabrikam, but **your personal account does have access**.

- ➊ [Sign in with your personal account](#)
- ➋ [Sign out and connect with another login](#)

[Service Status and Support](#) | [@AzureDevOps](#)

A highly specific 401 error case. In this case, both a personal Microsoft account and a work or school account (Azure AD) that have the same sign-in address exist. You've signed in with your work or school account, but your personal account is the identity with access to the organization.

Mitigation

In some cases, you might not know you have two identities with the same sign in address. The work or school Azure AD account might have been created by an administrator when you were added to Office365 or Azure AD.

To sign out of your current work or school Azure AD account, select **Sign in with your personal MSA account**, and then sign in by using your personal Microsoft account. After authentication, you should have access to the organization.

TIP

To avoid seeing this prompt, you can rename your Microsoft account. Then, only one identity (your work or school account, or Azure AD account) uses your sign-in address.

Scenario 2

Your personal Microsoft account doesn't have access, but your Azure AD account does. This scenario is an opposite version of the 401 error page. In this case, the personal account (Microsoft account identity) doesn't have access to the organization and the work or school account (Azure AD identity) does. The same guidance from Scenario 1 applies, but in reverse.

401 - Work or school, or Personal account

401

NOT AUTHORIZED

jamal@fabrikam.com has multiple accounts associated with it.

Your personal account does not have access to dev.azure.com/Fabrikam, but **your work or school account does have access**.

- ➊ [Sign in with your work or school account](#)
- ➋ [Sign out and connect with another login](#)

[Service Status and Support](#) | [@AzureDevOps](#)

Mitigation

If you enter your credentials correctly, but are redirected back to the original sign-in page, we recommend clearing all cookies, and then reattempting to sign in. If that doesn't fix the issue, contact customer support.

Troubleshoot TFS connectivity

Here's a list of the most frequently reported connection problems and what to do about them. Complete the list in the order indicated.

1. Verify that you have the required permissions.

If the errors that you receive indicate read-only or blocked actions, you might not have permissions to act on the data.

2. Verify that your computer is connected to the network and that it can access network resources.

3. Verify that TFS hasn't been taken offline. Talk with your TFS administrator.

4. Check whether your project has been moved to another project collection in TFS. If it has been moved, you must create a connection to the new server name.

For additional troubleshooting tips, see [TF31002: Unable to connect to this Team Foundation Server](#).

Switch organizations

When you use two or more organizations that are linked to Azure AD, such as organizations created in the Azure portal, the sign-out function might not work as expected. For example, you can't switch between different organizations to connect to multiple organizations that are linked to directory tenants.

When this problem occurs, a blank screen flashes several times. Then, one of the following error messages appears after you connect to or add a new connection in the **Connect to Team Foundation Server** dialog box:

TF31003: Either you have not entered the necessary credentials, or your user account does not have permission to connect to the Team Foundation Server

TF31002: Unable to connect to this Team Foundation Server

To resolve this issue, apply Visual Studio 2013.2 or install a later version from the [Visual Studio download website](#).

Another solution is to delete your browser cookies. For more information, see the support article [You can't switch between different organizations in Visual Studio Online](#).

Connect to TFS with Secure Sockets Layer

If you connect to a TFS instance that has Secure Sockets Layer (SSL) configured, you must install a certificate and clear the client cache. For details, see [Set up HTTPS with Secure Sockets Layer \(SSL\) for TFS - Configuring client computers](#).

Clear the cache on client computers

When the on-premises TFS configuration changes, such as when you move or split a project collection, you may need to clear the cache.

1. Sign in to your client computer for TFS by using the credentials of the user whose cache you want to clear.
2. Close any open instances of Visual Studio.
3. Open a browser and go to one of the following folders, depending on the operating system that's running on your computer:
 - **Windows 10** *Drive:\Users<i>UserName\AppData\Local\Microsoft\Team Foundation\6.0\Cache*
 - **Windows 8** *Drive:\Users<i>UserName\AppData\Local\Microsoft\Team Foundation\4.0\Cache*
 - **Windows 7 or Windows Vista** *Drive:\Users<i>UserName\AppData\Local\Microsoft\Team*

Foundation\2.0\Cache

4. Delete the contents of the Cache directory, including all subfolders.

TF31002: Unable to connect

8/1/2019 • 5 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

You might receive this error when you try to connect to Azure DevOps Services or an on-premises Azure DevOps Server from Visual Studio.

You receive this error when you try to connect to Azure DevOps Services

PROBLEM	RESOLUTION
You don't have an active account or license.	Check with your administrator that you're a member of the account and have an active, valid license. See Assign licenses to users for details.
Your Azure DevOps Services organization is connected to the Azure Active Directory.	When your Azure DevOps Services organization is connected to a directory that is associated with an Office 365 or Microsoft Azure subscription, only members in the directory can access the account. Check with your directory administrator to have them create an organizational account for you or add your account to the directory as external member .
You can't switch between different organizational accounts.	If you work with several organizations that connect to different directories, such as accounts created from the Microsoft Azure Portal, the sign-out function might not work as expected. For example, you can't switch between different organizational accounts to connect to multiple accounts that are linked to directory tenants. When this problem occurs, you see a flashing blank sign in dialog box several times. Then, you receive either TF31002 or TF31003 error after you connect to or add a new connection in "Connect to Team Foundation Server" dialog box. To resolve this problem, apply the most recent Visual Studio update . To learn more, see KB Article ID 2958966, You can't switch between different organizational accounts in Visual Studio Online .
You want to sign in to Azure DevOps Services from Visual Studio using different credentials.	See Connect to projects, Sign in with different credentials .

When you try to connect to an on-premises Azure DevOps Server from your client computer

If you determine that you're receiving this error from one computer but not others, or others aren't receiving this error, then check the problem resolutions that are outlined below.

PROBLEM	RESOLUTION
Your password has expired.	Verify that you entered your user ID and password correctly, and that your password hasn't expired.
You've entered an incorrect server URL.	Verify that you've entered the server URL correctly including the server name, port number, and protocol (http/https). See Connect to projects to learn more.
The TFS configuration has changed.	If the configuration for the on-premises Azure DevOps Server has changed, you must create a new connection. You might also need to clear the client cache .
You work remotely and need to connect to a TFS Proxy server to check in files to Team Foundation version control.	Configure Visual Studio to connect to TFS Proxy .
You're connecting to a later version of TFS than your Visual Studio client version.	Your version of Visual Studio or Team Explorer might be incompatible with Team Foundation Server. You might need to install one or more GDR packs. See Requirements and compatibility for details.
Your firewall is blocking TFS services.	See Allow a program to communicate through Windows Firewall .
Visual Studio stops responding when you run a query in Visual Studio.	Your computer might be configured to bypass the proxy server. Verify the configuration of the BypassProxyOnLocal setting on your computer. For more information, see BypassProxyOnLocal Configuration .

Several users can't connect to an on-premises Azure DevOps Server

If the problem occurs on more than one computer, contact your administrator to confirm whether the server is operational and available on the network.

As an administrator, check the event logs for the application-tier server to try to pinpoint the problem. Also, you can use the following table to determine whether the server is misconfigured. In the table, problems that are more likely to occur appear first. Try the resolutions in the order in which they appear, which increases the chance that you can solve the problem quickly.

PROBLEM	RESOLUTION
The <i>TFSService</i> account password has expired or is incorrect.	Many services for Team Foundation Server will stop running when the service account for Team Foundation has expired. For more information, see Change the service account or password for Team Foundation Server .
The application-tier server for Team Foundation is unavailable.	Verify whether each required service is running. If a required service isn't running, you must restart it. If necessary, set it to start automatically. For more information, see Stop and start services, application pools, and websites .
The network is unavailable.	Verify whether your network is operational.
A website identity for Team Foundation is configured incorrectly.	Verify or correct the server binding assignments that are made to websites for Team Foundation.

PROBLEM	RESOLUTION
Access to a website for Team Foundation has been restricted.	Verify or correct restrictions that are made to those websites that are based on IP addresses and domain names.
The firewall or ports are configured incorrectly.	Verify or correct port binding assignments for websites and port assignments for the firewall. First, you should open the administration console for Team Foundation, display the Application Tier page, and review the URL assignments. If necessary, you can click Change URL to modify the URL of a website. Next, you should verify the port assignments for Internet Information Services (IIS) and the ports that are allowed through the firewall. For more information, see Review Server Status and Settings and Verify or Correct Port Assignments .
Trust relationships between domains aren't configured correctly.	If a group of users can't access Team Foundation Server, you might have trust issues between domains.
When users connect to different versions of TFS from Visual Studio, for example, they connect to TFS 2012 and then TFS 2008, they can get the TF31002 error.	<p>This error can occur because the GUIDs for the TFS 2012 collection are the same as TFS 2008. The local client cache gets confused because it tries to maintain the same GUID-based local cache for both the 2008 server and the new Project Collection in 2012.</p> <p>To fix, run the TFSCconfig ChangeServerID command. See TFSCconfig ChangeServerID command.</p>

If the previous resolutions don't solve the problem, go to the [MSDN Forums - Visual Studio Team System —Team Foundation Server - Administration](#).

Troubleshoot network connections and lists of allowed addresses

9/3/2019 • 2 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

If you are having network connection issues to Azure DevOps, using NuGet, or connecting from Visual Studio 2015 and later versions, it may be because your security appliances are blocking connections now that Visual Studio uses TLS 1.2.

To fix this issue, update the security appliances in order to support TLS 1.2 for the following connections:

List of URLs for sign-in and licensing connections

- <https://management.core.windows.net>
- <https://login.microsoftonline.com>
- <https://login.live.com>
- <https://go.microsoft.com>
- <https://graph.windows.net>
- <https://app.vssps.dev.azure.com>
- <https://app.vssps.visualstudio.com>

A more general list of URLs for signing in to Azure DevOps and Azure

- <https://windows.net>
- <https://microsoftonline.com>
- <https://visualstudio.com>
- <https://microsoft.com>
- <https://live.com>
- <https://dev.azure.com>
- <https://management.core.windows.net>
- <https://app.vssps.dev.azure.com>
- <https://app.vssps.visualstudio.com>
- <https://vstsagentpackage.azureedge.net>
- <https://cdn.vsassets.io> -- hosts our CDN content
- <https://gallerycdn.vsassets.io> -- hosts Azure DevOps extensions
- <https://static2.sharepointonline.com> -- hosts some resources that we use in "office fabric" UI kit (fonts, etc)
- https://*.vstmrblob.vsassets.io -- hosts our TCM log data

IP range restrictions

To ensure your organization works with any existing firewall or IP restrictions, ensure that dev.azure.com and *dev.azure.com are open and update your allow-listed IPs to include the following IP addresses, based on your IP version. If you're currently allow-listing the 13.107.6.183 and 13.107.9.183 IP addresses, leave them in place, as you don't need to remove them.

IPv4 ranges

- `13.107.6.0/24`
- `13.107.9.0/24`
- `13.107.42.0/24`
- `13.107.43.0/24`

IPv6 ranges

- `2620:1ec:4::/48`
- `2620:1ec:a92::/48`
- `2620:1ec:21::/48`
- `2620:1ec:22::/48`

NuGet connections

- `https://azurewebsites.net`
- `https://nuget.org`

NOTE

Privately owned NuGet server URLs may not be included in the list above. You can check the NuGet servers you are using by opening up `%APPData%\Nuget\NuGet.Config`.

Get support and provide feedback

8/1/2019 • 4 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

IMPORTANT

Having problems? We can help. We offer a [live chat](#) (English only) support option. Choose from **Technical Support, Sales Support, Visual Studio (For your Company)**, and **Account, Subscription, and Billing Support**. Select your country from the dropdown menu, and then select **Live Chat (English)**.

Share your feedback and ideas with us, or join our communities. We're always working to improve Azure DevOps, and we want you to be part of the process!

Do you need to do any of the following?

- **Get advice** Visit StackOverflow for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Report a bug** Submit it through our Developer Community for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Suggest a feature or a fix** Submit your idea or issue through our Developer Community for [Azure DevOps Services](#) or [Azure DevOps Server](#).
- **Find out what's new in Azure DevOps** Check out the [current Azure DevOps Release Notes](#). These notes are updated every three weeks.

Azure DevOps product support

The primary support venues for Azure DevOps are as follows:

- [Azure DevOps Services Support](#)
- [Azure DevOps Server Support](#)

For technical support:

- [Basic support](#)
- [Premium support](#)

For billing support:

- [Azure DevOps](#)

Get live help

We offer a [live chat](#) (English only) support option. Choose from **Technical Support, Sales Support, Visual Studio (For your Company)**, and **Account, Subscription, and Billing Support**. Select your country from the dropdown menu, and then select **Live Chat (English)**.

Documentation feedback

All docs on docs.microsoft.com have a ratings tool in the lower right-hand corner of the page. It asks "Is this content helpful?" Answer **Yes** or **No** depending on your experience.

Add more detailed feedback by selecting the **Tell us more** link after selecting **Yes** or **No**. Check an appropriate box and enter what we can do to improve the content for you! Although we can't reply back, we collect and review this feedback regularly, and use your sentiments in our content planning.

Tips for effective feedback

If you just want to vent about the product or the docs, that's okay. It helps us a lot to know when you're happy or unhappy with an experience. For the most impact, though, provide details so we can better understand what we're doing right or wrong.

- Provide a little context. What problem were you trying to solve? At what point did it go wrong?
- What's your role? We don't need personal or professional details. Are you a dev? A manager? A business owner? When we understand our audience, we can come up with better solutions for you and other customers doing similar work.
- What version of the product were you using? What other products were you using with it?

The best feedback we get is clear and precise. For example:

- Product feedback: "I'm a project manager for a small start-up. I'm using Azure DevOps. I'm trying to create work item templates through the UI, but my changes don't seem to persist. It's not clear what I'm doing wrong."
- Doc feedback: "I'm a dev in a large organization that works on Java apps. I tried to use Maven with your build system in Azure DevOps Server 2017 Update 1 (15.112.26307.0), but I couldn't get the configuration shown in the docs to work."

The more details, the better!

What platform/version am I using?

You can tell what platform you use from the URL you use to connect to Azure DevOps Services, Azure DevOps Server, or Team Foundation Server.

Azure DevOps Services

An Azure DevOps URL consists of an organization name and dev.azure.com, for example:

`https://dev.azure.com/{yourorganization} .`

To learn the version number, enter the following address in a web browser:

`https://dev.azure.com/{yourorganization}/_home/About`

A page similar to the following example opens showing the version number.



About Azure DevOps Service

Version Dev17.M149.1 (build: AzureDevOps_M149_20190326.1)

© Microsoft Corporation. All rights reserved.

Azure DevOps Server

An on-premises URL consists of a server name, port number, and collection name, for example:

`https://ServerName:8080/tfs/CollectionName`

To learn the version number, enter the following address in a web browser:

https://ServerName:8080/tfs/_home/About

A page similar to the following example opens showing the version number.

The screenshot shows the 'About Microsoft Visual Studio Team Foundation Server' page. At the top, it says 'Visual Studio Team Foundation Server 2015'. Below that, it displays the title 'Microsoft Visual Studio Team Foundation Server'. Underneath the title, it shows 'Version 14.102.25423.0' and '© Microsoft Corporation. All rights reserved.'

ON-PREMISES RELEASE	UPDATE	VERSION NUMBER
Azure DevOps Server 2019	RTW	17.143.28511.3 (Dev17.M143)
Azure DevOps Server 2018	2018.3	16.131.28106.2
	2018.2	16.131.27701.1
	2018.1	16.122.27409.2
	RTW	16.122.27102.1
	RC2	16.122.26918.3
	RC1	16.121.26818.0
Azure DevOps Server 2017	Update 3	15.117.27024.0
	Update 3 RC	15.117.26912.0
	Update 2	15.117.26714.0
	Update 1	15.112.26307.0
	RTW	15.105.25910.0
	RC1	15.103.25603.0
Azure DevOps Server 2015	Update 3	14.102.25423.0
	Update 2.1	14.95.25229.0
	Update 2	14.95.25122.0
	Update 2 RC 2	14.95.25029.0
	Update 2 RC 1	14.95.25005.0
	Update 1	14.0.24712.0

ON-PREMISES RELEASE	UPDATE	VERSION NUMBER
	Update 1 RC 2	14.0.24626.0
	Update 1 RC 1	14.0.24606.0
	RTM	14.0.23128.0
	RC2	14.0.23102.0
	RC	14.0.22824.0
	CTP	14.0.22604.0
Azure DevOps Server 2013	Update 5	12.0.40629.0
	Update 4	12.0.31101.0
	Update 4 RC	12.0.31010.0
	Update 3	12.0.30723.0
	Update 3 RC	12.0.30626.0
	Update 2	12.0.30324.0
	RTM	12.0.21005.1
	RC	12.0.20827.3
Azure DevOps Server 2012	Update 4	11.0.61030.0
	Update 3	11.0.60610.1
	Update 2	11.0.60315.1
	CU 1	11.0.60123.100
	Update 1	11.0.51106.1
	RTM	11.0.50727.1
Azure DevOps Server 2010	CU 2	10.0.40219.371
	SP1	10.0.40219.1
	RTM	10.0.30319.1
Azure DevOps Server 2008	SP1	9.0.30729.1
	RTM	9.0.21022.8

ON-PREMISES RELEASE	UPDATE	VERSION NUMBER
Azure DevOps Server 2005	SP1	8.0.50727.762
	RTM	8.0.50727.147

Related articles

- [Azure DevOps features timeline](#)
- [Report a problem with Visual Studio](#)

Web portal navigation in Azure DevOps

8/1/2019 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

The web portal for Azure DevOps is organized around a set of services, as well as administrative pages and several task-specific features such as the search box. The service labels differ depending on whether you work from Azure DevOps Services or Azure DevOps on-premises and it's version.

IMPORTANT

Make sure that you select the correct version of this article for Azure DevOps Services or Azure DevOps Server, renamed from Team Foundation Server (TFS). The version selector is located above the table of contents.

Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

The web portal for Azure DevOps Server is organized around a set of services—such as, **Overview, Boards, Repos, Pipelines, Test Plans, and Artifacts**—as well as administrative pages and several task-specific features such as the search box. Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

The web portal for Team Foundation Server (TFS) is organized around a set of applications—such as, **Dashboards, Code, Work, Build and Release**—as well as administrative pages and several task-specific features such as the search box. Each service provides you with one or more pages which support a number of features and functional tasks. Within a page, you may then have a choice of options to select a specific artifact or add an artifact.

Here's what you need to know to get up and running using the web portal.

- **Open a service, page, or settings:** use to switch to a different service or functional area
- **Add an artifact or team:** use to quickly add a work item, Git repo, build or release pipelines, or a new team
- **Open another project or repo:** use to switch to a different project or access work items and pull requests defined in different projects, or items you've favorited
- **Open team artifacts, use breadcrumbs, selectors and directories:** use to navigate within a service, to open other artifacts or return to a root function
- **Work with favorites:** favorite artifacts to support quick navigation
- **Search box:** use to find code, work items, or wiki content
- **Your profile menu:** use to set personal preferences, notifications, and enable preview features
- **Settings:** use to add teams, manage security, and configure other project and organization-level resources.
- **Open a service, page, or settings:** use to switch to a different service or functional area
- **Add an artifact or team:** use to quickly add a work item, Git repo, build or release pipelines, or a new team
- **Open another project or repo, or switch to a different team:** use to switch to a different project or browse teams

- **Work across projects:** use to quickly open work assigned to you, your active pull requests, or items you've favorited
- **Open team artifacts, use breadcrumbs & selectors:** use to navigate within a service, to open other artifacts or return to a root function
- **Work with favorites:** favorite artifacts to support quick navigation
- **Search box:** use to find code, work items, or wiki content
- **Your profile menu:** use to set personal preferences, notifications, and enable preview features
- **Settings:** use to add teams, manage security, and configure other project and organization-level resources.

NOTE

Only those services that are enabled will appear in the user interface. For example, if **Boards** is disabled, then **Boards** or **Work** and all pages associated with that service won't appear. To enable or disable a service, see [Turn an Azure DevOps service on or off](#).

You select services—such as **Boards**, **Repos**, and **Pipelines**—from the sidebar and pages within those services.

ID	Title	State	Title
390	Cancel order form	Commit...	New
492	Build Settings Experience	Commit...	Committed
375	Check service status	Commit...	Check service status
543	Develop form	To Do	Develop form
372	Auto-save	To Do	To Do
539	Standardize	In Progr...	Auto-save

You select a service—such as **Code**, **Work**, and **Build and Release**—from the horizontal bar and pages within those services.

The screenshot shows the Azure DevOps Web Overview page. On the left, there are two cards: 'All items' (Work items: 56) and 'All bugs' (Work items: 0). The main area displays 'Work assigned to Jamal Hartnett (15)' with categories: Task (7), Product Backlog Item (4), and Other (4). A table lists work items with columns for ID, State, and Title.

ID	State	Title
390	● Commit...	⌚ Cancel order form
492	● New	🏆 Build Settings Experience
375	● Commit...	⌚ Check service status
543	● To Do	📅 Develop form
372	● To Do	📅 Auto-save
539	● In Progr...	📅 Standardize

Now that you have an understanding of how the user interface is structured, it's time to get started using it. As you can see, there are a lot of features and functionality.

If all you need is a code repository and bug tracking solution, then start with the [Get started with Git](#) and [Manage bugs](#).

To start planning and tracking work, see [About Agile tools](#).

Connect to the web portal, user accounts and licensing

You connect to the web portal through a supported web browser—such as the latest versions of Edge, Chrome, Safari, or Firefox. Only users who have been [added to a project](#) can connect. This is typically done by the organization owner.

Five account users are free as are Visual Studio subscribers and stakeholders. After that, you need to [pay for more users](#). Find out more about licensing from [Azure DevOps pricing](#).

Limited access is available to an unlimited number of stakeholders for free. For details, see [Work as a Stakeholder](#).

You connect to the web portal through a supported web browser—such as the latest versions of Edge, Chrome, Safari, or Firefox. Only users who have been [added to a project](#) can connect. This is typically done by a member of the Project Administrators group.

Limited access is available to an unlimited number of stakeholders for free. For details, see [Work as a Stakeholder](#). Most regular contributors must have a TFS client access license (CAL). All Visual Studio subscriptions include a TFS CAL. Find out more about licensing from [TFS pricing](#).

Refresh the web portal

If data doesn't appear as expected, the first thing to try is to refresh your web browser. Refreshing your client updates the local cache with changes that were made in another client or the server. To refresh the page or object you're currently viewing, refresh the page or choose the **Refresh** icon if available.

To avoid potential errors, you should refresh your client application under the following circumstances:

- Process changes are made
- Work item type definitions are added, removed, renamed or updated
- Area or iteration paths are added, removed, renamed or updated
- Users are added to or removed from security groups or permissions are updated
- A team member adds a new shared query or changes the name of a shared query
- A build definition is added or deleted
- A team or project is added or deleted

Differences between the web portal and Visual Studio

Although you can access source code, work items, and builds from both clients, some task-specific tools are only supported in the web browser or an IDE, but not in both.

WEB PORTAL	VISUAL STUDIO
<ul style="list-style-type: none">• Product backlog, Portfolio backlogs, Sprint backlogs, Task boards, Capacity planning• Kanban board• Dashboards, Widgets, and Charts• Team rooms• Request feedback• Web-based Test Management• Administration pages to administer accounts, team projects, and teams	<ul style="list-style-type: none">• Task specific interfaces that integrate with Git and TFVC, such as:<ul style="list-style-type: none">◦ Git: Changes Branches Pull Requests Sync Work Items Builds◦ TFVC: My Work Pending Changes Source Control Explorer Work Items Builds• Greater integration with work items and Office-integration clients. You can open a work item or query result in an office supported client.

Resources

- [Manage projects](#)
- [Project & Organizational Settings](#)

Open a service, page, or settings

8/1/2019 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

The web portal for Azure DevOps provides support for software development teams to collaborate through the planning, development, and release cycles. You can manage source code, plan and track work, define builds, run tests, and manage releases.

This article shows you how to navigate to functional and administrative tasks available from the web portal. There are three levels of administrative tasks: team, project, and organization.

If you don't have a project yet, [create one](#). If you don't have access to the project, [get invited to the team](#).

This article shows you how to navigate to functional and administrative tasks available from the web portal. There are four levels of administrative tasks: team, project, collection, and server.

If you don't have a project yet, [create one](#). If you don't have access to the project, [get invited to the team](#).

Open a service or functional task page

Services support getting work done—managing code, planning and tracking work, defining and managing pipelines, creating and running tests, and so on.

NOTE

Only those services that are enabled will appear in the user interface. For example, if **Boards** is disabled, then **Boards** or **Work** and all pages associated with that service won't appear. To enable or disable a service, see [Turn an Azure DevOps service on or off](#).

You open a service by choosing the service from the sidebar and then selecting from the available pages.

For example, here we select **Boards>Backlogs**.

The screenshot shows the 'Backlogs' page in the Azure DevOps interface. On the left, there's a sidebar with icons for Overview, Boards, Work Items, Boards, Backlogs (which is selected), Sprints, Queries, Repos, and Pipelines. The main area has a 'Web' dropdown, a star icon, and a refresh icon. It includes buttons for 'New Work Item', 'Backlog items Board', and more. Below is a table with columns for Order, Work item Type, and Title. The table lists 8 backlog items:

Order	Work item Type	Title
1	Product Backlog item	Hello World Web Site
2	Bug	Slow response on information form
3	Product Backlog item	Change initial view
4	Product Backlog item	Interim save on long form
5	Bug	Canadian addresses don't display correctly
6	Product Backlog item	Hello World Web Site
7	Product Backlog item	GSP locator interface
8	Product Backlog item	Request support

Within the page you may select a specific view or artifact, such as a team backlog or choose another page.

You open a service by choosing it from the horizontal blue bar. Then, select from the available pages.

For example, here we select **Work>Work Items**.

The screenshot shows the top navigation bar of the Azure DevOps interface. It includes a logo, the project name 'Fabrikam', a dropdown arrow, and tabs for Dashboards, Code, Work (which is highlighted with a red box), Build and Release, Test, Wiki, and a gear icon. Below the navigation bar, there's a secondary set of links: 'Work Items*' (highlighted with a red box), 'Backlogs', and 'Queries*'. The 'Work Items*' link is underlined.

Open team settings

Select configurations are made to teams through the team settings pages. For an overview of all team settings, see [About user, team, project, and organization-level settings](#).

1. Choose **Project Settings**.

Summary - Overview X

← → C https://dev.azure.com/fabrikam/FabrikamFiber

Azure DevOps Fabrikam / FabrikamFiber / Overview / Summary

FabrikamFiber Private Invite

Overview Summary Dashboards Wiki Boards Repos Pipelines Test Plans Artifacts

Welcome to the project!

What service would you like to start with?

Boards Repos Pipelines Test Plans Artifacts

or manage your services

Project settings

2. Expand **Boards** and choose **Team configuration**.

General

Overview

Services

Teams

Security

Notifications

Service hooks

Dashboards

Boards

Project configuration

Team configuration

> Build and release

> Code

Backlogs

See only the backlogs your team manages.

Backlog navigation levels

- Epic Epics
- Feature Features
- Backlog item Backlog items

Working days

Capacity and burndown are based on the days your team works.

Select days

- Monday
- Tuesday
- Wednesday
- Thursday
- Friday
- Saturday
- Sunday

3. Choose one of the pages **General**, **Iterations**, **Areas**, or **Templates** to configure settings for the team. To learn more, see [Manage teams](#).
4. If you need to switch to a different team, use the team selector within the breadcrumbs.

Project Settings > Team configuration > **Web**

General

Overview

Services

Teams

Security

Notifications

Service hooks

Work

General Iterations

Backlogs

See only the backlog

Backlog navigation levels

- Epic Epics
- Feature Features
- Backlog item Backlog items

Phone (Fabrikam Fiber)

Voice (Fabrikam Fiber)

Web (Fabrikam Fiber)

Customer Service (Fabrikam Fiber)

Fabrikam Fiber Team (Fabrikam F...)

Management team (Fabrikam Fib...)

More teams

Email (Fabrikam Fiber)

5. To add a team administrator, add team members, or change the team profile, choose **Teams** from the vertical sidebar, and then choose the name of the team you want to configure.

You open team settings from the top navigation bar. Select the team you want and then choose the gear icon. To learn more about switching your team focus, see [Switch project, repository, team](#).

Screenshot of the Azure DevOps interface showing the 'Work' tab selected. The 'Project settings' link in the bottom-left corner is highlighted with a red box.

Work

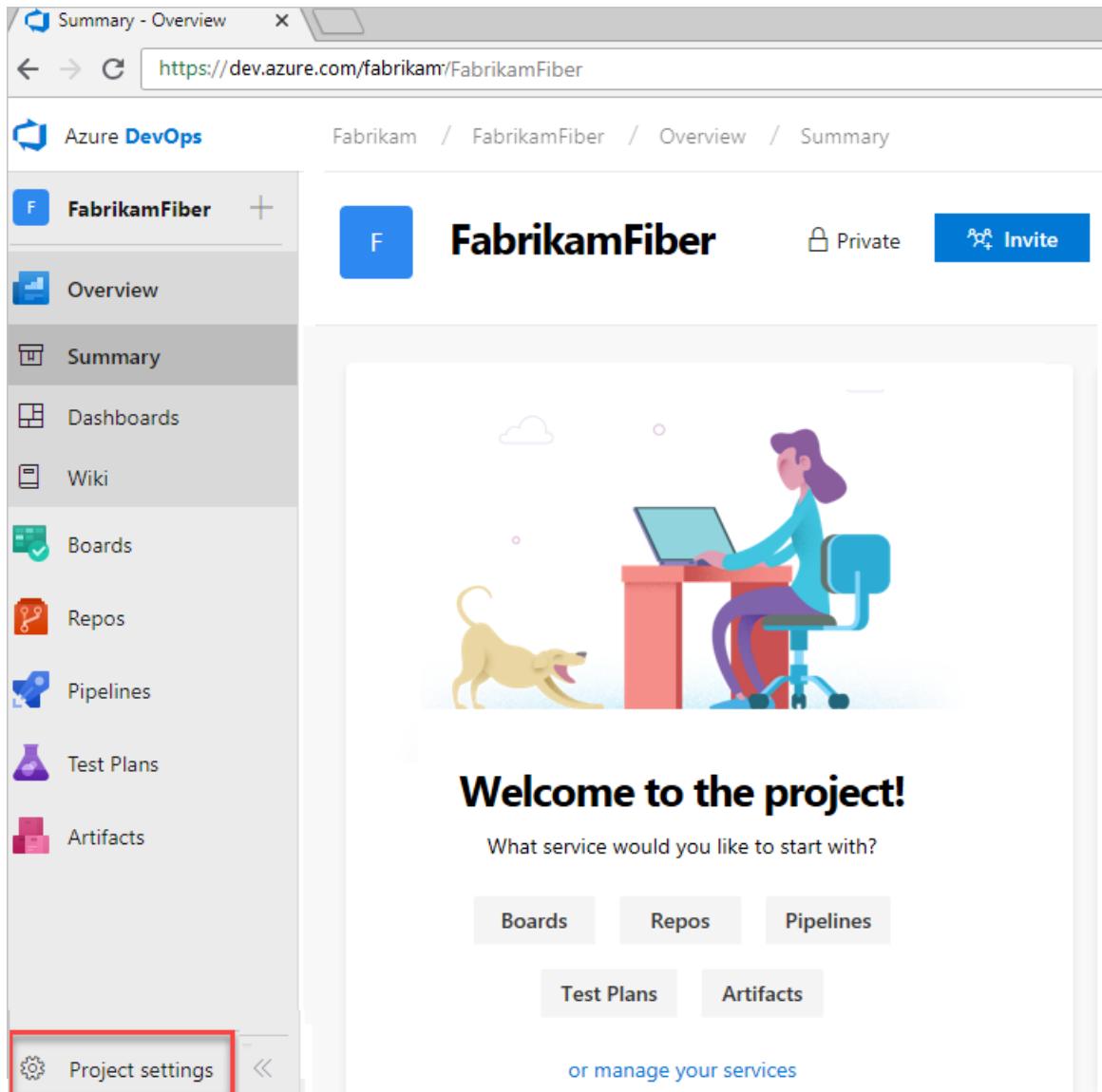
General Iterations Areas Templates

1. Choose one of the pages **General**, **Iterations**, **Areas**, or **Templates** to configure settings for the team. To learn more, see [Manage teams](#).
2. To add a team administrator, add team members, or change the team profile, choose **Overview**.
- 3.

Open project settings

Administrators configure resources for a project and manage project-level permissions from the **Project settings** pages. Tasks performed in this context can impact the project and team functions. For an overview of all project settings, see [Project administrator role and managing projects](#).

1. Choose **Project Settings**.



2. From there, you can choose a page from the list. Settings are organized based on the service they support. Expand or collapse the major sections such as **Boards**, **Build and release**, **Code**, **Test**, and **Extensions** to

select from the list.

The screenshot shows the 'Project Settings' interface. On the left, a sidebar lists various sections: General (selected), Overview, Services, Teams, Security, Notifications, Service hooks, Dashboards, Boards, Build and release, Code, Test, and Extensions. The 'General' section is expanded, showing sub-options like Overview, Services, Teams, Security, Notifications, Service hooks, and Dashboards. The 'Overview' option is selected and highlighted with a blue border. The main content area is titled 'Project details' with the sub-section 'Changes made here will affect all members and URLs associated with this project.' Below this, there's a large teal square placeholder image with the letters 'FF' in white. To the right of the image, there are fields for 'Name' (Fabrikam Fiber) with a 'Rename' button, and 'Description' (Web, voice, and phone apps). Further down, there are sections for 'Visibility' (Private, Edit) and 'Process' (MyScrum).

From a user context, open **Project settings** by choosing the gear icon.

Open any admin page by choosing its name. Choose or hover over the gear icon to access other administrative options. Note that you can choose any of the user-context areas—**Dashboards**, **Code**, **Work**—to return to the user context.

The screenshot shows the 'Files' section of the TFS 2017.2 interface. At the top, there's a navigation bar with 'Fabrikam Fiber / Fabrika...' and several tabs: 'Dashboards', 'Code', 'Work', 'Build and release', '...', and a gear icon. A red box highlights the gear icon. A red arrow points from the gear icon down to a dropdown menu. The dropdown menu lists various administrative options: 'Overview', 'Work', 'Security', 'Version Control', 'Policies', 'Agent Queues', 'Notifications', 'Service Hooks', 'Services', 'Test', 'Release', 'Dashboards', 'Project settings' (which is highlighted with a red box), and 'Organization settings'.

Open any admin page by choosing its name. Choose or hover over the gear icon to access other administrative options. Note that you can choose any of the user-context areas—**Home** or **Dashboards, Code, Work**—to return to the user context.

TFS 2017.2

The screenshot shows the 'Overview' page of TFS 2017.2. The top navigation bar includes 'Fabrikam Fiber', 'Home', 'Code', 'Work', 'Build & Release', 'Test', and a gear icon. Below the navigation is a horizontal menu with links: 'Overview', 'Work', 'Security', 'Version Control', 'Agent queues', 'Notifications', 'Service Hooks', 'Services', and 'Test'. A red box highlights the gear icon.

TFS 2017.1

The screenshot shows the 'Overview' page of TFS 2017.1. The top navigation bar includes 'Fabrikam Fiber', 'Dashboards', 'Code', 'Work', 'Build & Release', 'Test', and a gear icon. Below the navigation is a horizontal menu with links: 'Overview', 'Work', 'Security', 'Version Control', 'Agent Queues', 'Notifications', 'Service Hooks', 'Services', and 'Test'. A red box highlights the gear icon.

TFS 2017

The screenshot shows the 'Overview' page of TFS 2017. The top navigation bar includes 'Fabrikam Fiber', 'Home', 'Code', 'Work', 'Build & Release', 'Test', and a gear icon. Below the navigation is a horizontal menu with links: 'Overview', 'Work', 'Security', 'Alerts', 'Version Control', 'Agent Queues', 'Service Hooks', 'Services', and 'Test'. A red box highlights the gear icon.

Open Organization settings

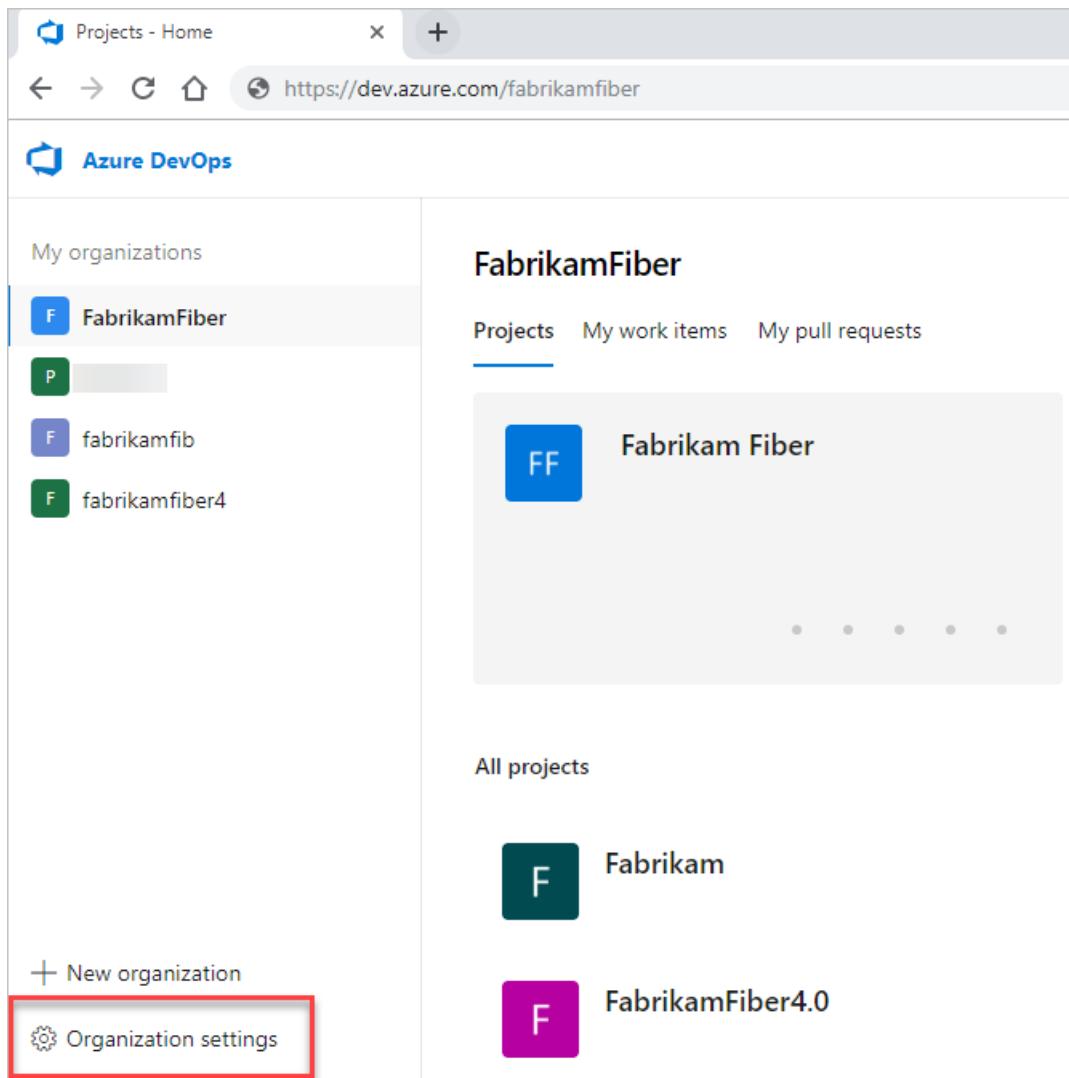
Organization owners and members of the Project Collection Administrators group configure resources for all projects or the entire organization, including adding users, from the Organization settings pages. This includes managing permissions at the organization-level. For an overview of all organization settings, see [Project collection administrator role and managing collections of projects](#).

Open Collection settings

Members of the Project Collection Administrators group configure resources for all projects or the entire project

collection from the Collection settings pages. This includes managing permissions at the collection-level. For an overview of all collection-level settings, see [Project collection administrator role and managing collections of projects](#).

1. Choose the  Azure DevOps logo to open **Projects**. Then choose **Admin settings**.



The screenshot shows the Azure DevOps Admin settings interface. On the left, there's a sidebar titled "My organizations" listing "FabrikamFiber" (selected), "fabrikamfib", and "fabrikamfiber4". Below the sidebar, there are buttons for "+ New organization" and "Organization settings", with "Organization settings" highlighted by a red box. The main area is titled "FabrikamFiber" and shows a project named "Fabrikam Fiber". Below this, under "All projects", are "Fabrikam" and "FabrikamFiber4.0". The URL in the browser bar is https://dev.azure.com/fabrikamfiber.

2. From there, you can choose a page from the list of settings. Settings are organized based on the service they support. Expand or collapse the major sections such as **Boards** and **Build and release** to select a page from the list.

Organization Settings > Projects

General

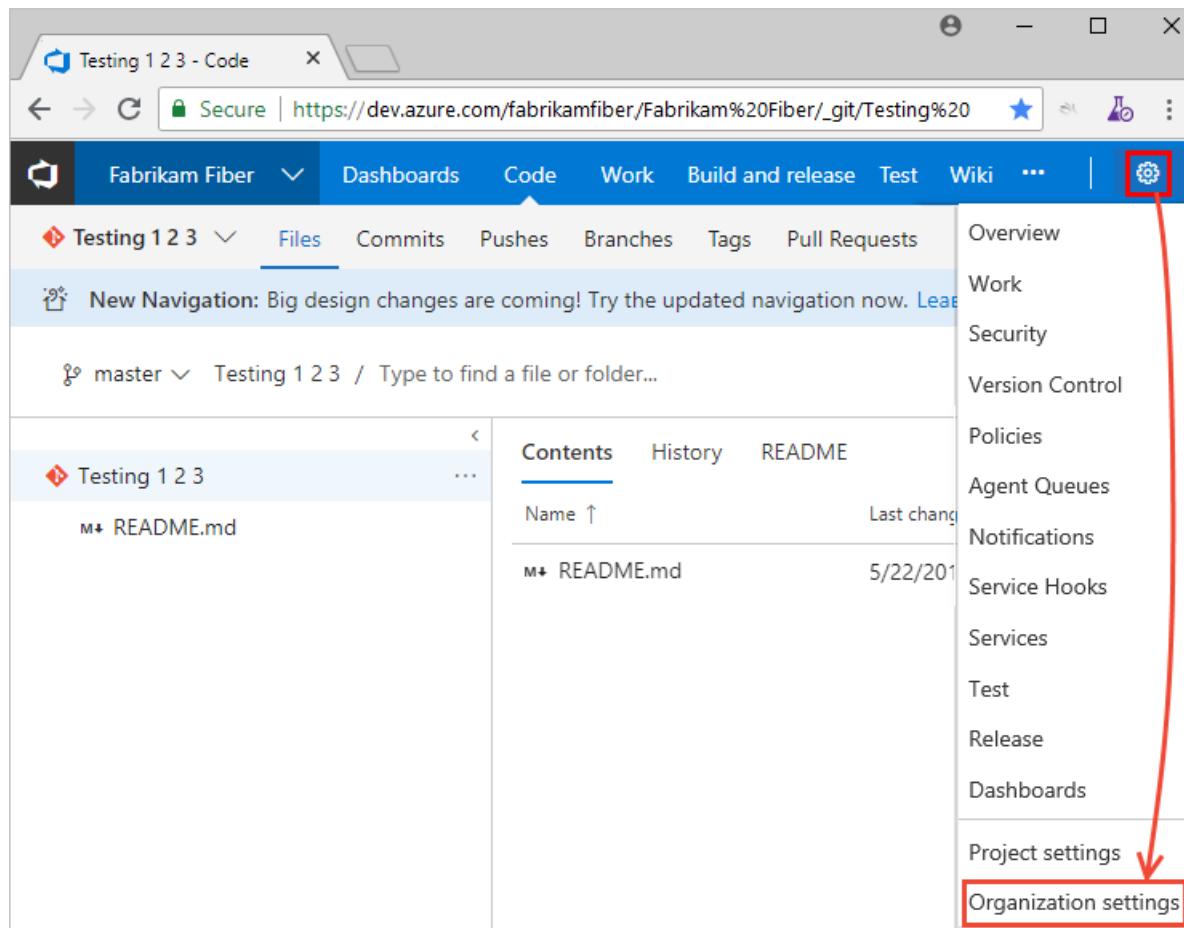
- Overview
- Projects**
- Policy
- Users
- Security
- Notifications
- Extensions
- Usage

> Boards

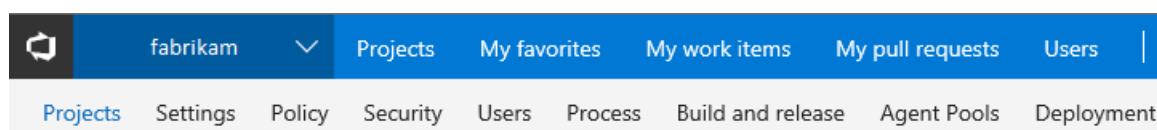
> Build and release

Project name	Process	Status	Description
Agile 11	MyAgile 2	Online	New agile project
Fabrikam Fiber	MyScrum	Online	Web, voice, and phone apps
Fabrikam Test	MyAgile Test	Online	MyAgile process customizations
Test Agile Repo	MyAgile	Online	
Visual Studio Code	Agile	Online	

- Choose the gear icon to open **Organization settings** or **Collection settings**.



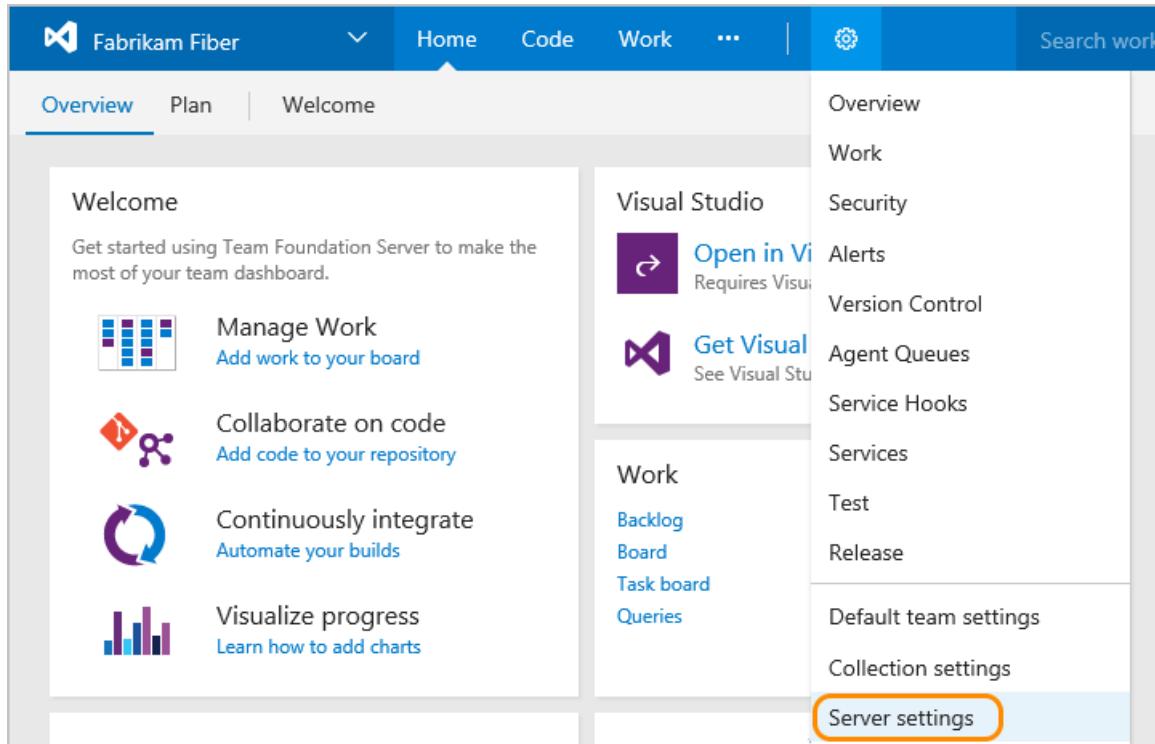
- From there, you can choose a page. Settings are organized based on the service they support.



Open Server settings

Members of the Team Foundation Server Administrators group configure resources for the server instance from the Server settings pages.

1. From the web portal home page for a project, choose or hover over the gear icon and select **Server settings**.



2. Choose **Access levels**, to set access levels for a member or group. For details, see [Change access levels](#).

If you don't see **Access levels**, you aren't a TFS administrator and don't have permission. [Here's how to get permissions.](#)

Related articles

- [Manage projects](#)
- [About team, project, and admin settings](#)

Add an artifact or team artifacts

9/27/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Select the service of interest to get started adding new artifacts or objects. For example, to add work items, choose **Boards** or **Work**. Some artifacts—such as a product backlog, Kanban board, portfolio backlogs—are added when you add a team.

Prior to adding an artifact, make sure that you've [selected the project and repository](#) that you want to work in.

Add work items, queries, or other work tracking artifacts

You can quickly add a query or work item when working from a **Boards** or **Work** page.

Choose a **Boards** page—such as **Work Items**, **Boards**, or **Backlogs**. Then choose the **+** plus icon and select from the menu of options.

The screenshot shows the 'Work Items' page in the Azure DevOps interface. On the left, there's a sidebar with links: Overview, Boards, Work Items (which is selected and highlighted in blue), Boards, Backlogs, and Sprints. The main area shows a list of work items with columns for ID, Title, and State. A red arrow points from the 'New work item' link in the top navigation bar down to a context menu that has appeared. This menu includes options like 'New query', 'New work item' (which is selected and highlighted in blue), 'Open in Queries', and a 'Types' dropdown.

From a **Work** page, you can add a work item from the menu of options as shown in the following image.

The screenshot shows the 'Work' page for the 'Fabrikam Fiber' project. The top navigation bar includes links for Fabrikam Fiber, Dashboards, Code, Work (which is selected and highlighted in blue), Build & Release, Test, and Wiki*. The main content area displays information about the 'Fabrikam Fiber' team and its support for various voice services. On the right side, there's a context menu for adding a new work item. The 'New Work Item' option is selected and highlighted in blue. Below it, a list of work item types is shown: Bug (selected), Epic, Feature, Issue, Task, Test Case, and User Story. The 'User Story' option is highlighted with a red box.

Or, you can open one of the pages—**Boards**, **Backlogs**, **Queries**, or **Plans**—to add an artifact specific to each of these functional pages.

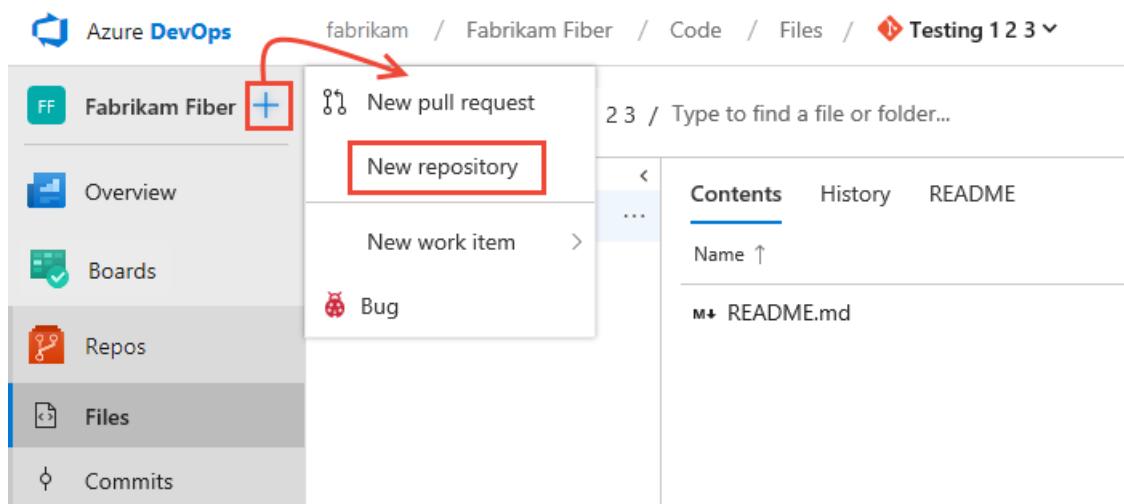
To add other work tracking artifacts, see one of the following articles:

- To add a board, backlog, or sprint backlog, first [add a team](#) which will be associated with those artifacts
- [Add a delivery plan](#)
- [Add a managed work item query](#)
- [Add work items](#).

Add a pull request or Git repository

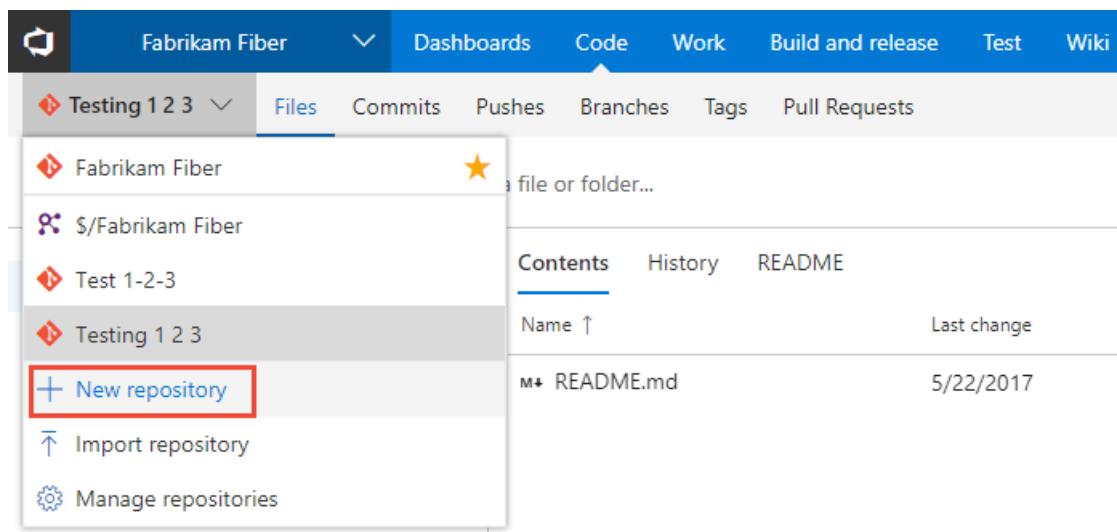
You can quickly add a pull request, Git repository, or work item using the **Add** menu when working from **Code**.

Expand the **Repos** service and choose **Files**, **Commits**, or **Pull Requests** (Git repos) or **Files**, **Changesets**, or **Shelvesets** (TFVC). Then, choose the **+** plus icon and select from the menu of options.



For details on adding a Git repository, see [Git repository](#).

From **Code**, open the context menu for the current repository and choose **+** **New repository**. For details on adding a Git repository, see [Git repository](#)



From one of the other **Code** pages, you can add files or folders, a new branch, or a new pull request.

Note that you can only add one TFVC repository per project, but an unlimited number of Git repositories. To learn more about Git artifacts, see one of the following articles:

- [Git repository](#)
- [Git branch](#)
- [Git pull request](#)
- [Add work items](#)

Add build and release pipelines

Expand **Pipelines** and choose **Builds** or **Releases**. Then choose the **+** plus icon and select from the menu of options.

The screenshot shows the Azure DevOps interface for the project 'Fabrikam Fiber'. The left sidebar has 'Pipelines' selected. The main area shows a summary for 'Fabrikam Fiber-Cl' with a 'Builds' tab highlighted. A red box surrounds the '+' icon in the top navigation bar, and a red arrow points to it from the text above. The interface includes a search bar for 'Build ID or build number' and tabs for 'Queued' and 'XAML'.

From **Build and Release**, choose **Builds**, **Releases**, or other page to add an artifact associated with that page.

The screenshot shows the 'Build and Release' interface for the 'Fabrikam Fiber' project. The top navigation bar includes 'Fabrikam Fiber', 'Build and Release', and a search bar. Below the navigation bar, there is a menu with tabs: 'Builds' (selected), 'Releases', 'Library', 'Task Groups', 'Deployment Groups*', and 'Build Tags'.

To learn more about adding other pipeline related artifacts, see the following articles:

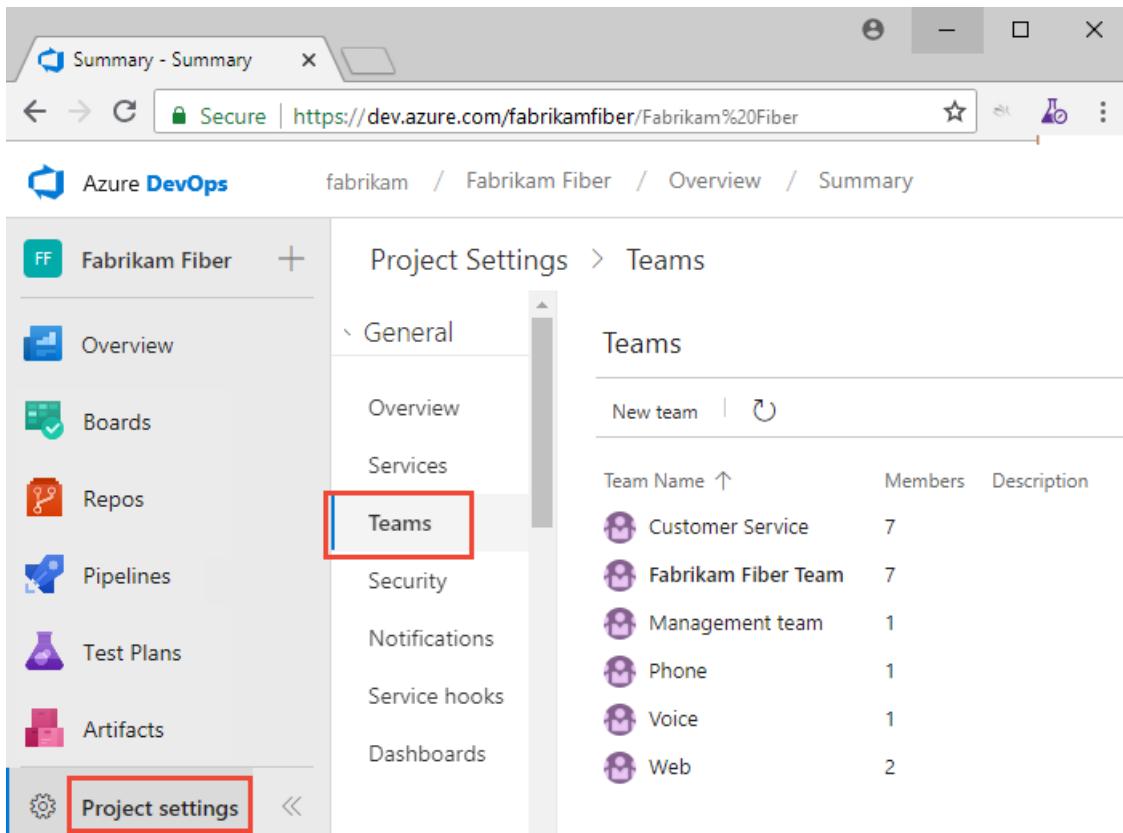
- [Deployment groups](#)
- [Task groups](#)
- [Variable groups](#)
- [Secure files](#)

Add a team

Agile tools and dashboards are typically associated with teams. You add teams to a project. To learn more about teams, see [About teams and Agile tools](#). To add a team, see [Add a team and team members](#).

View teams already defined

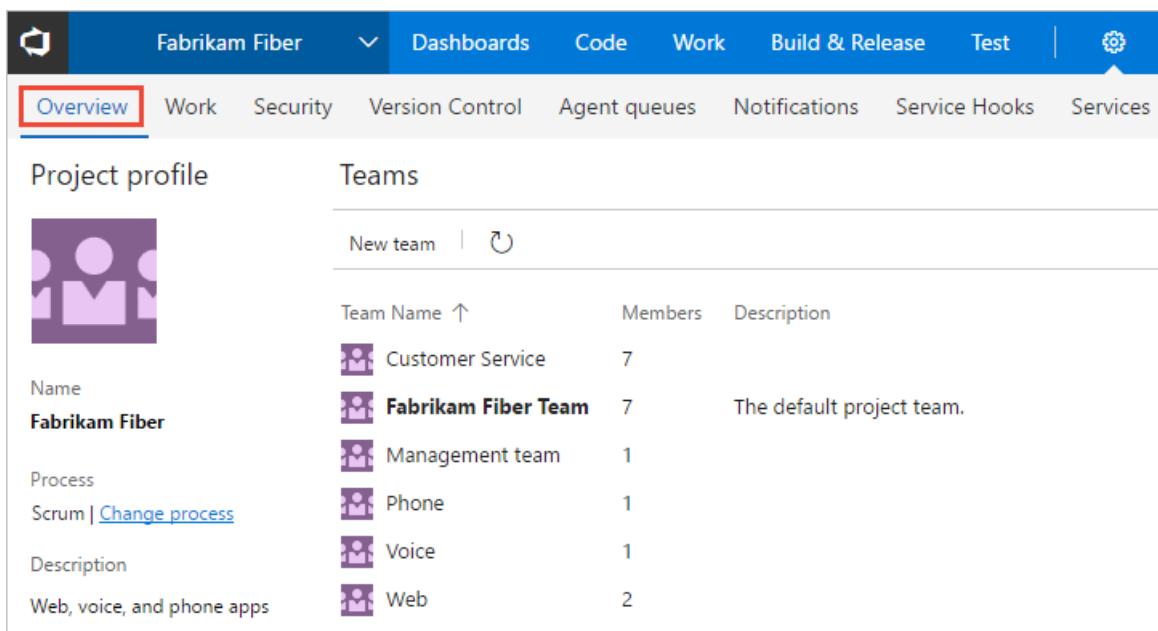
To view the set of defined teams, open **Project settings**, and choose **Overview**.



The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. On the left, there's a navigation bar with icons for Overview, Boards, Repos, Pipelines, Test Plans, Artifacts, and Project settings. The 'Project settings' icon is highlighted with a red box. A secondary navigation menu on the left lists 'General', 'Teams' (which is also highlighted with a red box), 'Security', 'Notifications', 'Service hooks', and 'Dashboards'. The main content area is titled 'Project Settings > Teams' and displays a table of teams with columns for Team Name, Members, and Description. The teams listed are Customer Service (7 members), Fabrikam Fiber Team (7 members, described as 'The default project team.'), Management team (1 member), Phone (1 member), Voice (1 member), and Web (2 members).

Team Name	Members	Description
Customer Service	7	
Fabrikam Fiber Team	7	The default project team.
Management team	1	
Phone	1	
Voice	1	
Web	2	

To view the set of defined teams, open the admin context for the project, and choose **Overview**.



The screenshot shows the 'Fabrikam Fiber' project overview page. The top navigation bar includes links for Overview, Work, Security, Version Control, Agent queues, Notifications, Service Hooks, and Services. The 'Overview' link is highlighted with a red box. On the left, there's a 'Project profile' section with details like Name (Fabrikam Fiber), Process (Scrum), and Description (Web, voice, and phone apps). The main content area is titled 'Teams' and displays a table of teams with columns for Team Name, Members, and Description. The teams listed are Customer Service (7 members), Fabrikam Fiber Team (7 members, described as 'The default project team.'), Management team (1 member), Phone (1 member), Voice (1 member), and Web (2 members).

Team Name	Members	Description
Customer Service	7	
Fabrikam Fiber Team	7	The default project team.
Management team	1	
Phone	1	
Voice	1	
Web	2	

Add a dashboard

Dashboards are associated with a team. Each team can create and configure a number of dashboards. To learn how, see [Add a dashboard](#).

Add a wiki

If you don't have a wiki yet, you can add one. Once added, you can add and update pages to that wiki.

- [Create a wiki](#)
- [Add and edit wiki pages](#)
- [Publish a Git repository to a wiki](#)

- [Create a wiki](#)
- [Add and edit wiki pages](#)

Related articles

- [Azure Artifacts](#)
- [Exploratory & Manual Testing](#)

Use breadcrumbs, selectors, and directories to navigate and open artifacts

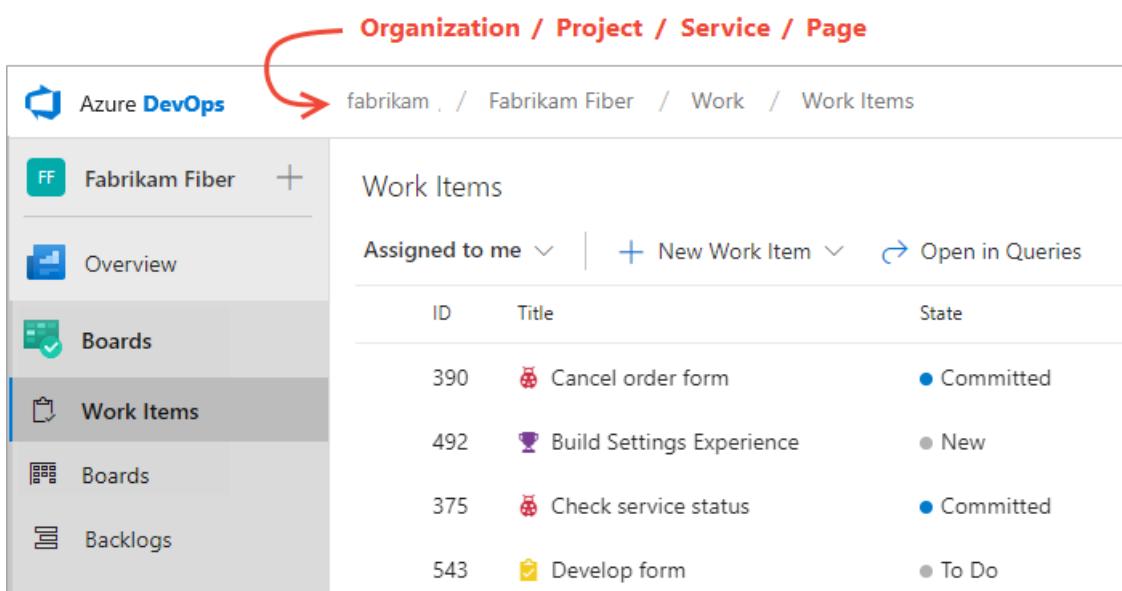
Use breadcrumbs and selectors to navigate and open artifacts

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#)

To quickly navigate to a feature or artifact—such as a dashboard, repository, product backlog, Kanban board, build pipeline—you can use breadcrumbs, selectors, and directories.

Organization and project breadcrumbs

To navigate to the project summary page, choose the project link within the breadcrumbs. To navigate to the organization page with all projects defined for the organization, choose the organization link.



The screenshot shows the Azure DevOps interface. At the top, a red curved arrow points from the word "Organization" in the breadcrumb "Organization / Project / Service / Page" down to the "Organization" link in the left sidebar. The sidebar has a dark grey header with the Azure DevOps logo and the text "Azure DevOps". Below this, there are several items: "Fabrikam Fiber" (selected), "Overview", "Boards", "Work Items" (selected), "Boards", and "Backlogs". The main content area has a light grey header "Work Items" with dropdowns for "Assigned to me", "New Work Item", and "Open in Queries". Below this is a table with columns "ID", "Title", and "State". The table contains four rows of work items:

ID	Title	State
390	Cancel order form	Committed
492	Build Settings Experience	New
375	Check service status	Committed
543	Develop form	To Do

Horizontal navigation doesn't provide a breadcrumb structure for the organization and project levels. Instead, you can select a recent team or project from the project/team selector.

A screenshot of the Microsoft DevOps interface. At the top, there's a navigation bar with links for Dashboards, Code, Work, Build & Release, and Test. Below the navigation bar is a sidebar for the 'Fabrikam Fiber' project. The sidebar includes links for 'Fabrikam Fiber Home', 'Recent projects/teams' (with 'Agile 11' and 'FabrikamFiber' listed), 'Fabrikam Fiber A' (which is selected and highlighted in blue), 'Fabrikam Fiber PB', 'Browse...', and 'New team'. A note below the sidebar says: 'A README.md file is intended to quickly orient readers to what your project can do. Learn more about Markdown.' A red box highlights the dropdown arrow next to 'Fabrikam Fiber A'.

Choosing **Browse all** opens the **projects** page.

Selectors

Selectors are used to select an artifact within the current page. Most Agile tools are defined for a team and therefore require selection of the team artifact or tool.

Selectors are used to select an artifact within the current page. Most Agile tools are defined for a team and therefore require selection of the team as well as the specific page.

NOTE

When you navigate to a specific page or artifact, the system remembers your selection. You use selectors to choose a different artifact within the current page.

Example: Dashboard selector

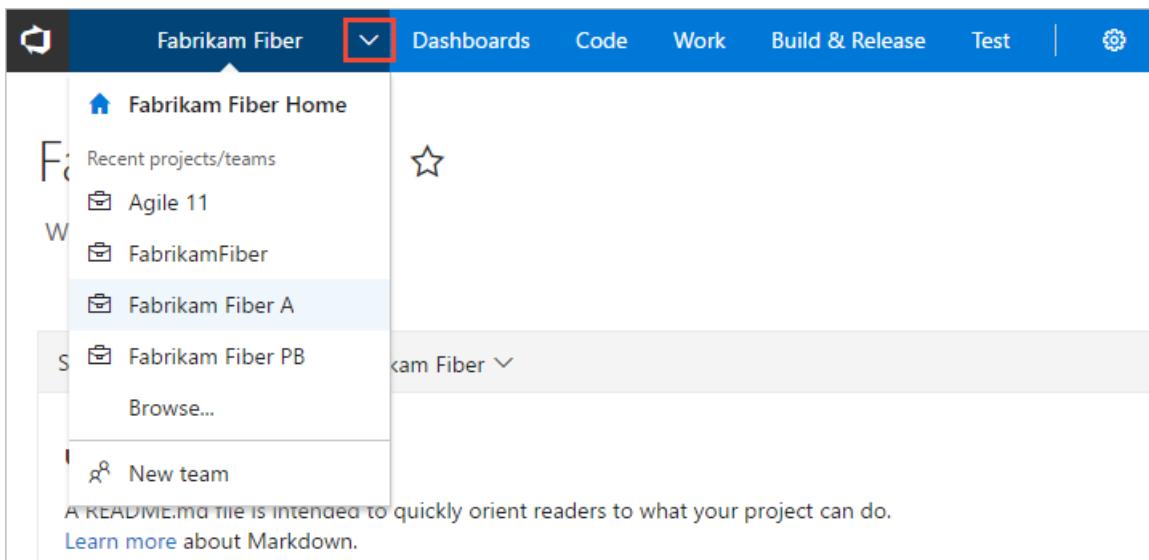
Within **Dashboards**, you open a specific dashboard from the selector.

A screenshot of the 'Dashboards' page. At the top, there are buttons for 'Edit' and 'Refresh'. Below that is a search bar labeled 'Search dashboards'. Underneath the search bar are two tabs: 'Mine' and 'All dashboards'. A list of dashboards follows, including 'Fabrikam Fiber Team Analytics', 'Web Overview' (which is selected and highlighted in grey), 'Web Bugs', 'Web Test', 'Web Work in Progress', 'Customer Service', 'Customer Service Overview', '+ New dashboard', and 'Browse all dashboards'. To the right of the dashboard list, there's a vertical sidebar with sections for 'Team (15)', '4 Other', 'order form', 'Settings Experience', 'service status', 'top form', 'Save', 'Gridize', 'Work Item', and 'title'. A red box highlights the dropdown arrow next to 'Web Overview'.

This particular selector features these navigational elements:

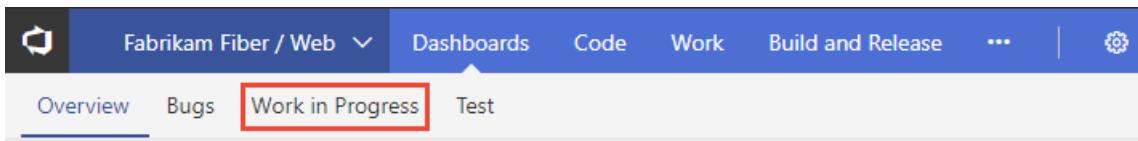
- Search box for filtering dashboards based on a team name or keyword
- Two pages you can choose from:
 - **Mine** (dashboards you created) which are organized by team
 - **All** (dashboards created by everyone) which are listed alphabetically
- Dashboards you've favorited will appear at the top of the selector
- **Add new dashboard** feature
- **Browse all dashboards** - opens **Dashboards>All**

Within **Dashboards**, you select the team whose dashboards you want to view.



Then, choose the name of the dashboard to view it.

For example, here we open the *Work in Progress* dashboard.



Example: Backlogs

From the **Boards>Backlogs** page, you use the selector to switch to another team's backlog. Again, favorited backlogs appear towards the top of the menu. You can also filter the list based on a team name or keyword.

The screenshot shows a search interface for team backlogs. On the left, a sidebar lists teams: Phone, Web, Customer Service, Fabrikam Fiber Team, Management team, and Voice. Below this is a link to 'Browse all team backlogs'. The main area displays a table of backlog items:

	Product Backlog	Request support	Cancel order form	State
8	Product Backlog	Request support		Committed
9	Product Backlog	Cancel order form		New

Or, choose **Browse all team backlogs** to open the **Backlogs>All** page.

(1) Select the team from the project/team selector, choose (2) **Work**, (3) **Backlogs**, and then (4) the product backlog, which is **Backlog items** (for Scrum), **Stories** (for Agile), or **Requirements** (for CMMI).

The screenshot shows the 'Backlogs>All' page. The navigation path is highlighted with red circles: (1) Project/Team selector (Fabrikam Fiber / Fabrika...), (2) Work tab, (3) Backlogs tab, and (4) Backlog items sub-tab. The main area displays the 'Product backlog' with tabs for 'Backlog' and 'Board'. At the bottom are buttons for 'New', 'Create query', 'Column options', and 'Email'.

To choose another team, open the project/team selector and select a different team or choose the **Browse** option.

The screenshot shows the project/team selector. It lists recent teams: Fabrikam Fiber / Phone, Fabrikam Fiber / Voice, Fabrikam Fiber / Web, and Fabrikam Fiber / Management team. Below this are links for 'Browse...' and 'New team'.

Artifact breadcrumbs and selectors

Within select pages, breadcrumbs are provided to support navigating within the page or opening an artifact.

Example: Queries folders and breadcrumbs

For example, when working in the **Queries** pages, you can navigate to a subfolder, folder, or page.

The screenshot shows the 'Queries' page in the Azure DevOps interface. The top navigation bar displays the path: 'Queries page / Queries folder / Queries subfolder / query'. On the left, there is a sidebar with links like 'Overview', 'Boards', 'Work Items', 'Boards', 'Backlogs', 'Sprints', and 'Queries'. The main content area shows a table of queries with columns: ID, State, Assigned To, Remaining Work, and Title. A red arrow points from the breadcrumb text to the 'Work in Progress' link in the selector menu.

Also, you can choose a query that you've favorited from the selector menu. Or, you can choose to browse all queries which returns you to the **All Queries** page.

The screenshot shows the 'All Queries' page. The top navigation bar displays the path: 'All Queries > Shared Queries > Current Sprint > Work in Progress'. Below the navigation, there is a table of queries. A red box highlights the 'Work in Progress' link in the selector menu, which is currently selected. A red arrow points from the breadcrumb text to this highlighted link.

Example: Pipeline folders and breadcrumbs

Breadcrumb-and-selector navigation elements are used within most services that support defining and organizing artifacts within folders. This includes **Pipelines** or **Build and Release** applications pages.

The screenshot shows the 'Build Definitions' page. The top navigation bar displays the path: 'Build Definitions / RedTeam / RedCreator'. Below the navigation, there is a table of build definitions. A red box highlights the 'RedCreator' link in the selector menu, which is currently selected. A red arrow points from the breadcrumb text to this highlighted link.

Choose the **Deployment** breadcrumb link to return to the *Deployment* folder.

Fabrikam Fiber > Dashboards > Code > Work > Build and release > Test > ...

Builds Releases Releases* Library Task groups Deployment Groups Build Tags

... > Deployment > Fabrikam Fiber-Cl

Directories

Directories provide a filterable list of all artifacts defined for a service area. Often times when you navigate to an application, it will open the application's directory.

For example, here is the **Boards>Boards** directory.

Boards

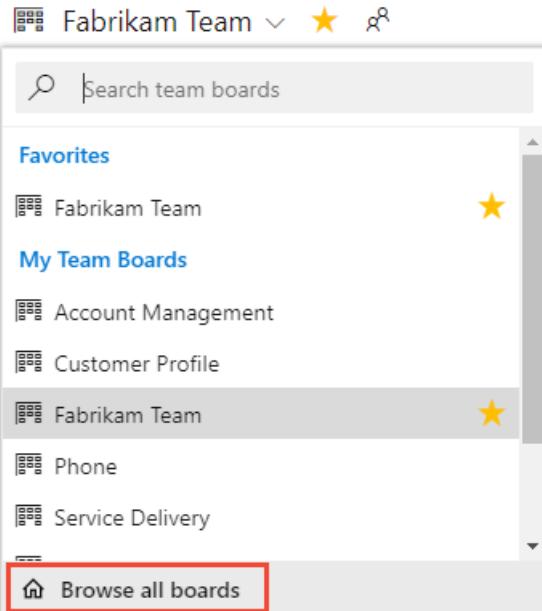
Name	Team
Continue where you left off	
Fabrikam Team boards	⭐ Fabrikam Team
My favorite boards (1)	
Fabrikam Team boards	⭐ Fabrikam Team
My team boards (8)	
Account Management boards	⭐ Account Management
Customer Profile boards	⭐ Customer Profile
Fabrikam Team boards	⭐ Fabrikam Team
Phone boards	⭐ Phone
Service Delivery boards	⭐ Service Delivery
Service Status boards	⭐ Service Status
Shopping Cart boards	⭐ Shopping Cart
TV boards	⭐ TV
All teams boards	

It lists boards in the following order:

- Your last visited board
- Your favorited boards
- All boards of teams that you belong to
- All boards defined for the project in alphabetical order.

Choose the filter icon to filter the list as described in [Filter basics](#).

From a specific page, you can open the directory from the breadcrumbs or a selector. For example, choose **Browse all boards** from the Boards selector.

OPEN FROM BREADCRUMBfabrikam / Fabrikam Fiber / Boards / **Boards****OPEN FROM SELECTOR**

Team profiles

- Open a team profile to quickly access items defined for a team. The team profile is available from the **Overview>Dashboards, Boards>Boards, Boards>Backlogs, and Boards>Sprints** pages.

The screenshot shows the 'Service Delivery' board selected in the 'Service Delivery' section of the page. A red box highlights the 'Service Delivery' board icon.

A panel opens that shows all items defined for the team.

The screenshot shows the 'Service Delivery' team profile panel. It displays a list of items including 'Service Delivery Boards', 'Service Delivery Backlogs', 'Sprint 2 Sprints', and 'Overview Dashboards', each with a yellow star icon. The 'Items' tab is selected.

- You can filter the list to show only **Dashboards**, **Boards**, **Backlogs**, or **Sprints** by choosing from the menu.

Service Delivery
Fabrikam Fiber
Team Settings

Items Members (7)

All Items

Dashboard
Boards
Backlogs
Sprints

- To view the team admins and members of the team, choose **Members**.

Service Delivery
Fabrikam Fiber
Team Settings

Items Members (7)

Admins

Members

	Cristina Potra
	Christie Church
	Chuck Reinhart
	Jamal Hartnett
	Johnnie McLeod
	Raisa Pokrovskaya

- To view or change the team configuration, choose **Team Settings**.

You can then add [team members](#), [team admins](#), or navigate to [team notifications](#), or team [iterations](#) and [area paths](#).

See also [Manage and configure team tools](#).

Related articles

- [About teams and Agile tools](#)

- [Add an artifact or team](#)
- [Set favorites](#)
- [Open a service or page](#)
- [Filter basics](#)

Switch project, repository, team

8/1/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

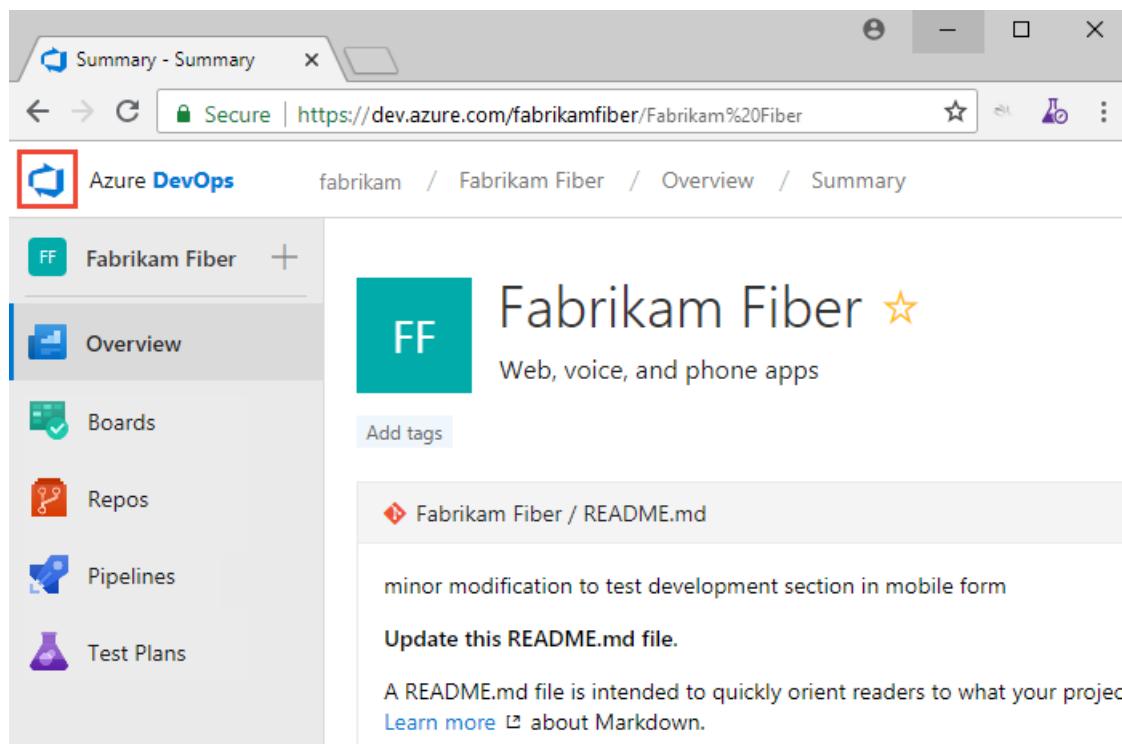
Several features depend on the project, repository, or team that you have selected. For example, dashboards, backlogs, and board views will change depending on the project and team you select.

Also, when you add a work item, the system references the default area and iteration paths defined for the team context. Work items you add from the team dashboard (new work item widget) and queries page are assigned the team default iteration. Work items you add from a team backlog or board, are assigned the team default backlog iteration. To learn more, see [About teams and Agile tools](#).

View and open a project

From the **Projects** page you can quickly navigate to a project that you have permissions to view.

1. Choose the  Azure DevOps logo to open **Projects**.



The projects you most recently viewed are displayed, followed by a list of all projects in alphabetic order.

2. Hover over the dots and you can open the service of interest for that project.

Projects

Filter projects

+ Create project

The screenshot shows a list of projects. At the top, there are two cards: 'Fabrikam Fiber' (Web, voice, and phone apps) and 'MyFirstProject'. Both cards have a small ellipsis icon at the bottom right.

All projects

The screenshot shows a list of projects. It includes 'Agile 11' (New agile project), 'Demo 11' (Agile team project), 'Fabrikam Fiber' (Web, voice, and phone apps), and 'MyFirstProject'. The 'Demo 11' card features four small icons: a green checkmark, a red document, a blue rocket, and a purple flask, all enclosed in a red rectangular border. Ellipsis icons are present at the bottom of each card.

3. You can filter the project and team list using the *Filter projects* search box. Simply type a keyword contained within the name of a project or team. Here we type **Fabrikam** to find all projects or teams with *Fabrikam* in their name.

Projects

Fabrikam X

+ Create project

The screenshot shows a list of projects filtered by 'Fabrikam'. It includes 'Fabrikam Fiber' (Web, voice, and phone apps), 'Fabrikam Test' (Project used to verify MyAgile process customizations), and 'FabrikamFiber' (Customer-focused apps under development based on Agile process). Ellipsis icons are present at the bottom of each card.

4. Choose **Create Project** to add a project. You must be an account administrator or a member of the Project Collection Administrators group to [add a project](#).

The screenshot shows the 'Projects' page in Azure DevOps. On the left, there's a sidebar titled 'My accounts' with three items: 'fabrikamfib' (dark blue icon), 'FabrikamFiber' (light blue icon, currently selected), and 'fabrikam-fiber' (green icon). The main area has a title 'Projects' with a 'Filter projects' search bar and a red-bordered 'Create project' button. Below this, there are two project cards: 'Fabrikam Fiber' (Web, voice, and phone apps) with a teal icon and 'MyFirstProject' with a pink icon. A horizontal ellipsis indicates more projects.

From the **Projects** page you can quickly navigate to a project or a team that you've accessed or worked in previously. Projects and teams are listed in the order you've last accessed, with the most recent five projects accessed appearing first. All projects you've accessed are listed within the **All** section.

1. Choose the Azure DevOps logo to open **Projects**.

The screenshot shows a browser window with the title 'Summary - Overview'. The address bar shows a secure connection to 'https://dev.azure.com/fabrikamfiber/Fabrikam%20Fiber'. The navigation bar includes back, forward, refresh, and search buttons, along with a star icon and other icons. The main menu bar at the top has links for 'Fabrikam Fiber', 'Dashboards', 'Code', 'Work', 'Build and release', 'Test', and 'Wiki'. The 'Fabrikam Fiber' link is highlighted with a red box.

The projects you most recently viewed are displayed, followed by a list of all projects in alphabetic order.

The screenshot shows the 'Projects' page. At the top, there are tabs for 'Projects', 'My favorites', 'My work items', 'My pull requests', and '...'. A 'New Project' button is also at the top right. Below this, a 'Recent' section lists three projects: 'Fabrikam Fiber' (with a star icon), 'Fabrikam Fiber / Web' (with a star icon), and 'FabrikamFiber'. There's also a 'Filter projects and teams' search bar.

2. As you hover over a project or team, you can choose one of the links to go to **Home** or **Dashboards**, **Code**, **Work**, **Build and Release**, **Test**, or **Wiki** pages. Choose the star icon to mark the project as a favorite.

The screenshot shows a browser window with the 'Fabrikam Fiber' link in the address bar highlighted with a red box. The navigation bar below includes 'Dashboards', 'Code', 'Work', 'Build and Release', 'Test', and 'Wiki' links. A yellow star icon is located at the end of the navigation bar.

3. You can filter the project and team list using the *Filter projects and teams* search box. Simply type a keyword contained within the name of a project or team. Here we type **Fabrikam** to find all projects or teams with *Fabrikam* in their name.

Projects

Fabrikam 

Results

 Fabrikam Fiber

 Fabrikam Fiber / Customer Service 

 Fabrikam Fiber / Management team 

 Fabrikam Fiber / Phone 

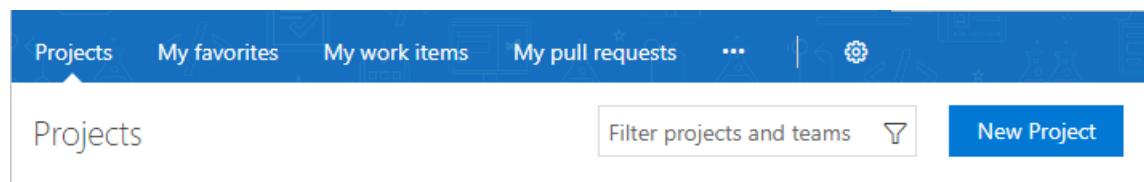
 Fabrikam Fiber / Voice 

 Fabrikam Fiber / Web 

 Fabrikam Test

 FabrikamFiber

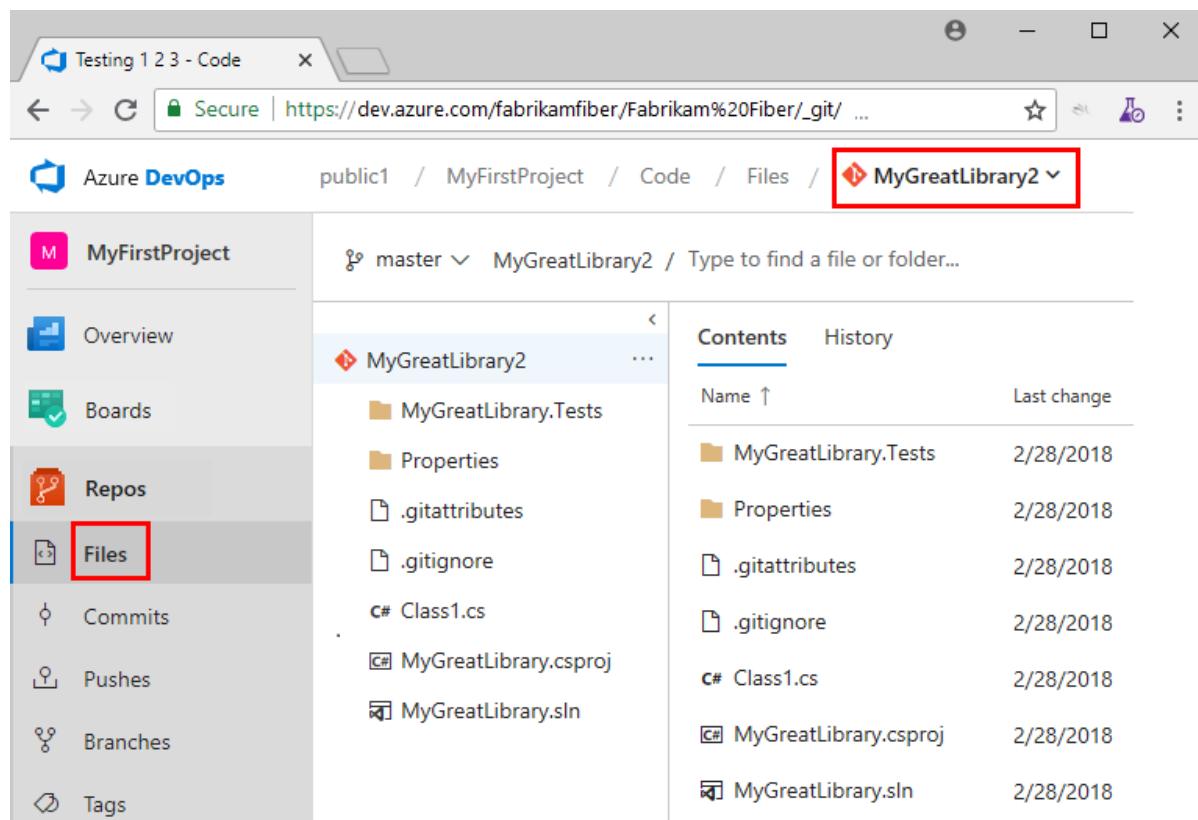
4. Choose **New Project** to add a project. You must be an account administrator or a member of the Project Collection Administrators group to [add a project](#).



The screenshot shows the Azure DevOps interface for managing projects. At the top, there's a navigation bar with links for 'Projects', 'My favorites', 'My work items', 'My pull requests', and more. Below the navigation is a search bar labeled 'Filter projects and teams' with a magnifying glass icon. A prominent blue button on the right is labeled 'New Project'. The main area displays a list of projects under the heading 'Projects'.

View and open a repository

1. Choose **Repos>Files**.



The screenshot shows the Azure DevOps repository interface. On the left, a sidebar menu has the 'Files' tab selected, indicated by a red box. The main area shows a file tree for a repository named 'MyGreatLibrary2'. The repository structure includes 'MyGreatLibrary2', 'MyGreatLibrary.Tests', 'Properties', '.gitattributes', '.gitignore', 'Class1.cs', 'MyGreatLibrary.csproj', and 'MyGreatLibrary.sln'. To the right, a table lists these files with columns for 'Name' and 'Last change'. The entire interface is framed by a red border.

Name	Last change
MyGreatLibrary.Tests	2/28/2018
Properties	2/28/2018
.gitattributes	2/28/2018
.gitignore	2/28/2018
Class1.cs	2/28/2018
MyGreatLibrary.csproj	2/28/2018
MyGreatLibrary.sln	2/28/2018

2. Select the repository of interest from the repository selector.

The screenshot shows a user interface for selecting a repository. At the top, there's a breadcrumb navigation: 'public1 / MyFirstProject / Code / Files /'. Below this is a dropdown menu labeled 'MyGreatLibrary2' with a red box around it. A red arrow points from this box down to a list of repositories. The list includes 'breadth-of-trees', 'Compose', 'GVFS', 'MyGreatLibrary', 'MyGreatLibrary2' (which is highlighted with a yellow star icon and has a red box around it), 'RedBot', 'Specs', 'VSCode', and several options at the bottom: '+ New repository', '↑ Import repository', and 'Manage repositories'. To the right of the list, there's a column for 'Commits' with entries like '2fb94e4' repeated multiple times.

1. Choose **Code**.

The screenshot shows the main project page for 'MyFirstProject'. At the top, there's a navigation bar with tabs: 'MyFirstProject' (with a bell icon), 'Dashboards', 'Code' (which is highlighted with a red box), 'Work', 'Build and Release', and 'Wiki'. The main content area features a large pink square with a white 'M' logo. To its right, the project name 'MyFirstProject' is displayed in large text, with the subtitle 'No description provided' below it. Underneath, there's a button labeled 'TypeScript'. At the bottom of the page, there's a section titled 'VSCode / README.md'.

2. Select the repository from the selector.

The screenshot shows the Azure DevOps repository interface. At the top, there's a navigation bar with tabs: 'Files' (highlighted with a red box and circle '1'), 'Commits', 'Pushes', 'Branches', 'Tags', and 'Pull Requests'. To the right of the tabs is a 'Clone' button. Below the navigation bar is a search bar with the placeholder 'Filter repositories' and a dropdown menu with the option 'a file or folder...'. To the right of the search bar is a green checkmark icon followed by the word 'succeeded'. The main area displays a list of repositories on the left and a detailed history table on the right. The repository 'MyGreatLibrary' is highlighted with a red box and circle '2'.

	History	Last change	Commits	
tLibrary.Tests	2/28/2018	2bfb94e4	create library	
es	2/28/2018	2bfb94e4	create library	
outes	2/28/2018	2bfb94e4	create library	
re	5/2/2018	573ba15d	Updated .gitig	
s	2/28/2018	2bfb94e4	create library	
MyGreatLibrary.csproj	2/28/2018	2bfb94e4	create library	
MyGreatLibrary.sln	2/28/2018	2bfb94e4	create library	

Switch to a different team

From a user page, one under—**Boards**, **Repos**, **Pipelines**, or **Test Plans**—you can't switch to a different team, you can [only select team artifacts](#).

From a **Project Settings>Work>Team configuration** page, you select a team from the team selector breadcrumb.

The screenshot shows the 'Project Settings > Team configuration' page. The breadcrumb at the top indicates the current selection is 'Fabrikam Fiber Team'. On the left, there's a sidebar with sections like 'General', 'Boards', and 'Teams'. The 'General' section is expanded, showing 'Overview', 'Services', 'Teams', 'Security', 'Notifications', 'Service hooks', and 'Dashboards'. The 'Teams' section is also expanded, showing 'Fabrikam Fiber Team' (which is highlighted with a red box). In the main content area, there's a 'Work' section with tabs for 'General' (selected) and 'Iterations'. Below that is a 'Backlogs' section with a 'Backlog navigation' dropdown and checkboxes for 'Epic', 'Feature', and 'Backlog'. There's also a 'Working days' section with a note about capacity and burndown, and a 'Select days' section where 'Monday' and 'Tuesday' are checked. A dropdown menu for selecting a team is open, listing 'Phone (Fabrikam Fiber)', 'Voice (Fabrikam Fiber)', 'Web (Fabrikam Fiber)', 'Customer Service (Fabrikam Fiber)', 'Fabrikam Fiber Team (Fabrikam Fib...', 'Management team (Fabrikam Fiber)', and 'Email (Fabrikam Fiber)'. The 'Fabrikam Fiber Team (Fabrikam Fib...' item is highlighted with a red box.

You can switch your team focus to one that you've recently viewed from the project/team selector. If you don't see the team or project you want, choose [Browse...](#) or choose the Azure DevOps logo to [access the Projects page](#).

The screenshot shows the Azure DevOps interface with the 'Fabrikam Fiber' project selected in the recent projects/teams dropdown. The dropdown menu includes options like 'Recent projects/teams', 'Agile 11', 'FabrikamFiber', 'Fabrikam Fiber A' (selected), 'Fabrikam Fiber PB', 'Browse...', 'New team', and a README.md file information.

Fabrikam Fiber Home

Recent projects/teams

Agile 11

W FabrikamFiber

W Fabrikam Fiber A

S Fabrikam Fiber PB

Browse...

New team

A README.md file is intended to quickly orient readers to what your project can do.
Learn more about Markdown.

TFS 2017.1 To switch your team focus to a project or team you've recently viewed, hover over the ![] (././_img/project-icon.png) Azure DevOps logo and choose from the drop-down menu of options. If you don't see the team or project you want, choose **Browse...** to [browse all projects and teams](work-across-projects.md).

The screenshot shows the TFS 2017.1 interface with the 'Fabrikam Fiber / Web' project selected in the recent projects/teams dropdown. The dropdown menu includes options like 'Recent projects/teams', 'Fabrikam Fiber / Web', 'Fabrikam Fiber', 'Fabrikam Fiber / Fabrikam Fiber Team', 'Fabrikam Fiber Scrum', 'Browse...', 'My Projects', 'My Favorites', 'My Work items', 'My Pull requests', 'More', 'New team', and 'New project'. A message 'new project!' is displayed next to the 'New project' option.

Fabrikam Fiber / Web

Recent projects/teams

Fabrikam Fiber / Web

Fabrikam Fiber

Fabrikam Fiber / Fabrikam Fiber Team

Fabrikam Fiber Scrum

Browse...

new project!

My Projects

My Favorites

My Work items

My Pull requests

More >

New team

+ New project

TFS 2017

Open the project/team drop-down menu and select the project/team that you've recently visited. If you don't see the team or project you want, choose **Browse all** to browse all projects and teams.

Fabrikam Fiber

Recent projects/teams

Fabrikam Fiber

Fabrikam Fiber / Fabrikam Fiber Team

Browse all

Server home

New team

New team project

Continuously integrate
Automate your builds

Visualize progress
Learn how to add charts

Open in Visual Studio
Requires Visual Studio 2013+

Get Visual Studio
See Visual Studio downloads

Work

Backlog

Board

Task board

Queries

Related articles

- [Work across projects](#)
- [Add teams](#)

Tutorial: Set personal or team favorites

8/1/2019 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Favorite  those views that you frequently access. You can favorite all sorts of Azure DevOps features and tools—such as a project, repository, build pipeline, dashboard, backlog, board, or query. You can set favorites for yourself or your team.

As your code base, work tracking efforts, developer operations, and organization grows, you'll want to be able to quickly navigate to those view of interest to you and your team. Setting favorites allows you to do just that.

Team favorites are a quick way for members of your team to quickly access shared resources of interest. You favorite an item for yourself by choosing the  star icon. The favorited item will then show up easily from one or more directory lists. You set favorites for a team through the context menu for the definition, view, or artifact.

In this tutorial you'll learn how to view your personal favorites and to favorite or unfavorite the following views:

- Project or team
- Dashboard
- Team backlog, board, shared query, or other Azure Boards view
- Repository
- Build and release definition
- Test plans

- Project
- Shared query
- Repository
- Build and release definition
- Test plans

Prerequisites

- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.
- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.

- You must connect to a project through the web portal. If you don't have a project yet, [create one](#). To connect to the web portal, see [Connect to a project](#).
- You must be a member of the **Contributors** or an administrators security group of the project. To get added, [Add users to a project or team](#).
- To favorite projects, backlogs, boards, queries, dashboards, or pipeline views, you must have **Stakeholder** access or higher.
- To favorite repositories, or delivery plans, you must have **Basic** access or higher.

- To favorite test plans, you must have **Basic + Test Plans** access level or equivalent.

For details about the different access levels, see [About access levels](#).

View personal favorites

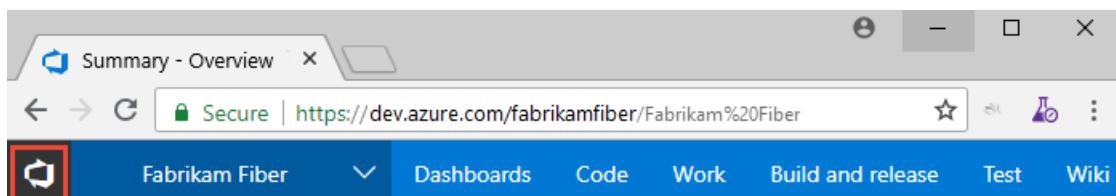
Access views that you have favorited by choosing the inbox icon, and then choosing **Favorites**.

Category	Item	Rating
Projects	Fabrikam Fiber	★
Teams	Phone	★
	Voice	★
	Web	★
Dashboards	Fabrikam Fiber Team Analytics	★
Plans	Backlog team plans	★
	Fabrikam Fiber Feature plans	★
Queries	All items	★
	All items on all projects	★
	Assigned to me	★

NOTE

If a service is disabled, then you can't favorite an artifact or view of that service. For example, if **Boards** is disabled, then the favorite groups—Plans, Boards, Backlogs, Analytics views, Sprints, and Queries and all Analytics widgets—are disabled. To re-enable a service, see [Turn an Azure DevOps service on or off](#).

- Access views that you have favorited by choosing the Azure DevOps logo to open **Projects**.



- Choose **My Favorites** to quickly access any view or item that you've marked as a favorite.

Favorites

Filter favorites



Queries

Bug Triage	Fabrikam Fiber	.../Shared Queries/Current Iteration	
My Bugs	Contoso	Shared Queries	
Open User Stories	Contoso	.../Shared Queries/Current Iteration	
Product Planning	Fabrikam Fiber	Shared Queries	
Product Planning	Contoso	Shared Queries	

Favorite a project or team

1. To favorite a project, open the project **Summary** page and choose the star icon.

The screenshot shows the Azure DevOps interface for the 'Fabrikam Fiber' project. The left sidebar has a 'Summary' link highlighted with a red box. The main content area displays the project name 'Fabrikam Fiber' with a yellow star icon next to it. Below the project name, there is a 'Web, voice, and phone apps' section and a 'README.md' file entry.

2. To favorite a team artifact, open **Boards>Boards** or **Boards>Backlogs**. Select the team you want to favorite from the team selector and choose the star icon.

The screenshot shows the 'Boards > Backlogs' page. The 'Backlog items backlog' section is visible, and a yellow star icon is located next to the team selector in the top right corner.

3. To favorite other team artifacts, choose the team icon, and then choose the star icon next to one of the listed artifacts.

X

Phone
Fabrikam Fiber
Team Settings

Items Members (1)

All Items

Phone Boards ★

Phone Backlogs ★

Phone Sprints ★

Favorite a project

To favorite a project, open the project **Summary** page and choose the ★ star icon.

FabrikamFiber ★

Customer-focused apps under development based on Agile process.

Members K +

Activity

Code

Build & Rel No builds yet

Use continuous integration

Improve code quality by detecting breaking changes as soon as they happen.

Setup Build

Learn more about continuous integration

Or, you can favorite a project from the **Projects** page by choosing the ★ star icon next to the project.

Favorite a dashboard

- From **Overview>Dashboards**, open the selector and choose the **Browse all dashboards** option.

The screenshot shows the Microsoft Power BI 'Mine' page. On the left, there's a sidebar with a search bar at the top. Below it, under 'Favorites', are 'Fabrikam Team Analytics' and 'Fabrikam Fiber ...'. Under 'Account Management', is 'Account Management Overview'. Under 'Customer Profile', is 'Customer Profile Overview'. Under 'Fabrikam Team', is 'Fabrikam Team Analytics'. At the bottom of the sidebar is a red-bordered button labeled 'Browse all dashboards'. To the right, a preview of the 'Fabrikam Team Overview' dashboard is shown, featuring a green card with the number '6' and 'Items' below it, and a purple card with the number '0' and 'Commits in last 7 d...' below it. Below the preview, there's a section titled 'Work items by State'.

2. The **Mine** page shows your favorited dashboards, and all dashboards of teams that you belong to. The **All** page (shown below) lists all dashboards defined for the project in alphabetical order. You can filter the list by team or by keyword.

Dashboards

The screenshot shows the Microsoft Power BI 'All' page. At the top, there are tabs for 'Mine' (highlighted with a red box) and 'All' (highlighted with a red box). There's also a '+ New dashboard' button and a filter icon. Below the tabs, there's a 'Filter dashboards' button and a 'Filter by team' dropdown menu with a search bar and a 'Clear' button. The main area displays a list of dashboards:

Name	Team
Analytics	Fabrikam Team
Bug status	Fabrikam Team
Bugs	Internet
Overview	Account Management
Overview	Customer Profile
Overview	Email
Overview	Fabrikam Team
Overview	Internet
Overview	Phone
Overview	Service Delivery
Overview	Service Status
Team Guidance	Fabrikam Team
Work in Progress	Internet

TIP

You can change the sort order of the list by choosing the column label.

3. To favorite a dashboard, hover over the dashboard and choose the star icon.

The screenshot shows the Azure DevOps interface with the 'Analytics' dashboard selected. At the top, there is a search bar and a 'Team' dropdown. Below the dashboard name, there is a 'Add to favorites' button with a star icon, which is highlighted with a red box. The 'Fabrikam Team' section is also visible.

Favoriting a dashboard will cause it to appear on your **Favorites** page and towards the top in the **Dashboards** selection menu.

Favorite a team's backlog, Kanban board, or other view

You can favorite several Agile tools for a team from a **Boards** page.

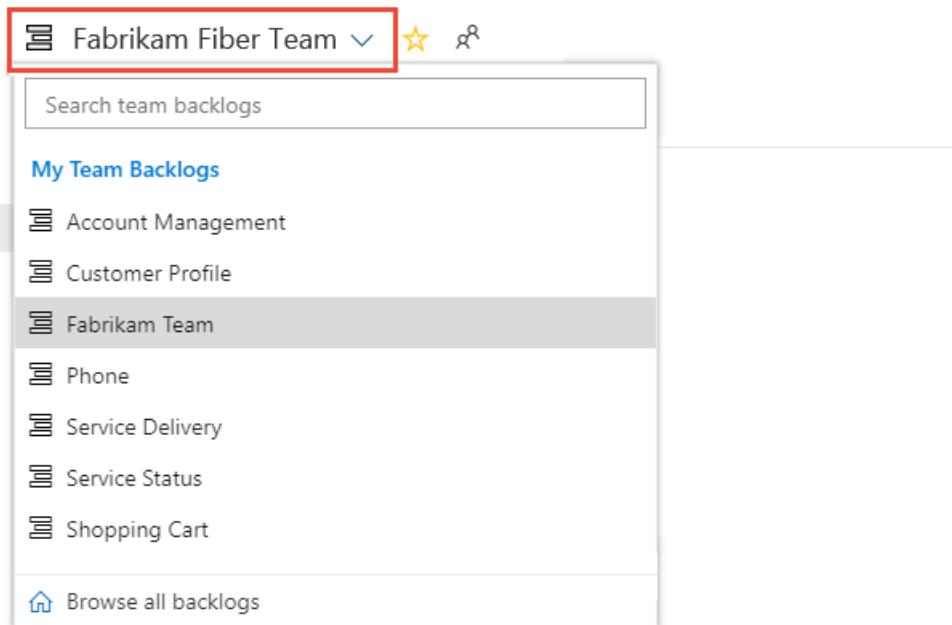
1. Choose **Boards**, and then choose the page of interest, such as **Boards**, **Backlogs**, or **Sprints**.

For example, here we choose (1) **Work** and then (2) **Backlogs**.

The screenshot shows the Azure DevOps 'Boards' page for the 'Fabrikam Fiber' project. The left sidebar has options like 'Overview', 'Boards', 'Work Items', 'Boards', 'Backlogs' (which is highlighted with a red box), 'Sprints', and 'Queries'. The main area shows the 'Fabrikam Fiber Team' backlog with 7 items. A 'New Work Item' button and a 'Backlog items Board' link are also present.

Order	Assigned To	State	Title
1	Jamal Hartnett	Committed	Hello World Web Site
2	Jamal Hartnett	Committed	Slow response on informa
3	Raisa Pokrovskaya	New	Add an information form
4	Raisa Pokrovskaya	New	Change initial view
5	Christie Church	Committed	Secure sign-in
6	Johnnie McLeod	Approved	Welcome back page
7	Christie Church	Committed	Cancel order form

To choose a specific team backlog, open the selector and select a different team or choose the **Browse all team backlogs** option. Or, you can enter a keyword in the search box to filter the list of team backlogs for the project.



2. Choose the star icon to favorite a team backlog. Favorited artifacts (favorited icon) appear on your **Favorites** page and towards the top of the team backlog selector menu.

Favorite a shared query

Open **Boards>Queries** and choose the **All** page. Expand a folder as needed. Choose the star icon next to the query you want to favorite.

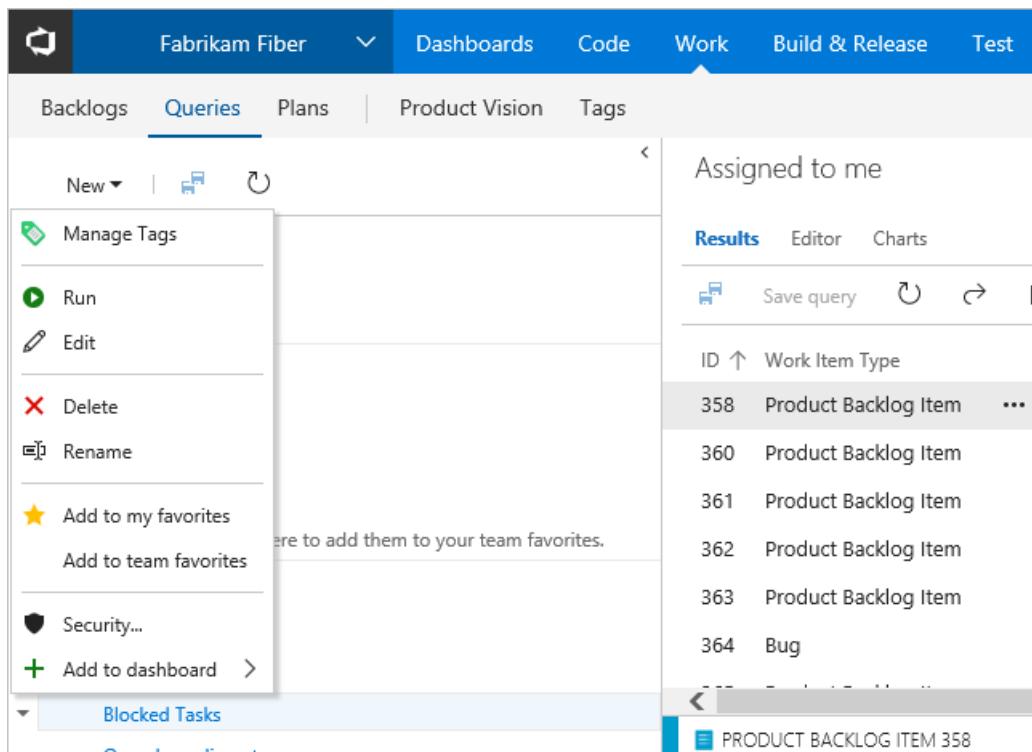
Or, open the context menu of the query, and then select **Add to Team Favorites**, and then select from the list of teams.

Queries

The screenshot shows the 'All' queries page with the 'Blocked Tasks' query selected. The context menu for this query is open, showing options like Run query, Edit, Rename, Delete, Add to Team Favorites, Security..., and Manage Tags. The 'Add to Team Favorites' option is highlighted with a red box. A dropdown menu lists several teams as favorite options: Customer Service, Fabrikam Fiber Team, Management team, and Phone. The 'Phone' option is also highlighted with a red box.

You can also set a query as a personal favorite by opening the query and choosing the star icon.

Open **Work>Queries**. Next, open the *** actions icon menu of the shared query you want to favorite, and then select **Add to my favorites** or **Add to team favorites**.



The screenshot shows the Microsoft Teams interface with the 'Fabrikam Fiber' team selected. The 'Queries' tab is active. A context menu is open on the 'Shared Queries' page, with the 'Add to my favorites' option highlighted. The main pane displays a list of queries under 'Assigned to me', including Product Backlog Item 358, 360, 361, 362, 363, and 364, along with a Bug item.

Favorite a delivery plan

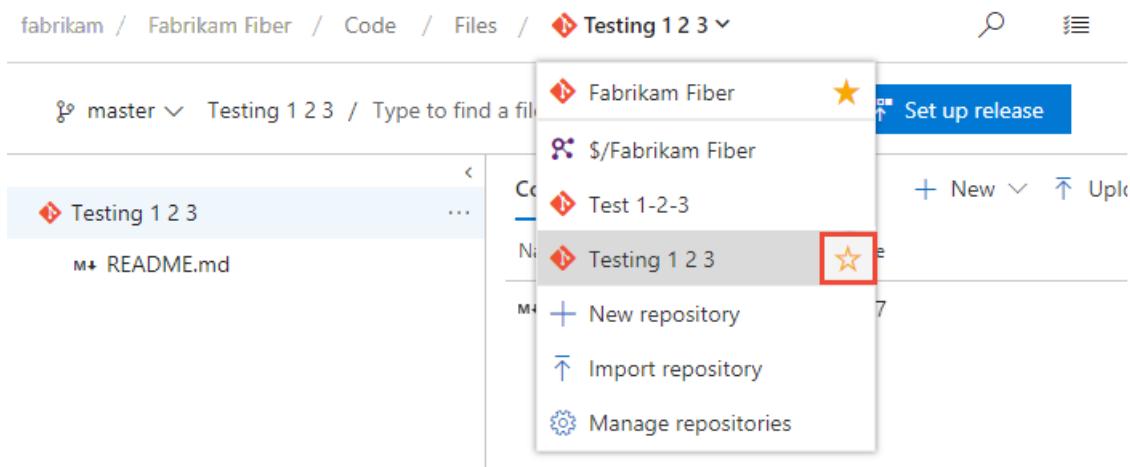
To learn more about delivery plans, see [Review team Delivery Plans](#).

To mark a delivery plan as a favorite, open the **Boards>Plans** page and choose the  star icon next to the Delivery Plan.

To mark a delivery plan as a favorite, open the **Work>Plans** page and choose the  star icon next to the Delivery Plan.

Favorite a repository

From any **Repos** page, open the repository selector and choose the  star icon for the repository you want to favorite.



The screenshot shows the Microsoft Teams interface with the 'Code' tab selected. A context menu is open on the 'Repos' page, with the 'Testing 1 2 3' repository selected and its star icon highlighted with a red box. The menu also includes options for 'Set up release', 'New', 'Import repository', and 'Manage repositories'.

From any **Code** page, open the repository selector and choose the star icon next to the repository you want to favorite.

The screenshot shows the Azure DevOps Code interface. At the top, there's a navigation bar with 'Fabrikam Fiber' selected. Below it, a sidebar on the left lists repositories: '\$/Fabrikam Fiber', 'Fabrikam Fiber' (which has a yellow star icon highlighted with a red box), 'Testing 1 2 3', '+ New repository', '+ Import repository', and 'Manage repositories'. The main area shows a list of files under 'Fabrikam Fiber': 'page-1.md' (modified on 10/15/2015), 'page-2.md' (modified on 10/15/2015), 'page-3.md' (modified on 9/21/2016), and 'README.md' (modified on 5/19/2017). There are tabs for 'Contents', 'History', and 'README'.

Favorite a build pipeline

Open **Pipelines>Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to my favorites** or **Add to team favorites**.

The screenshot shows the Azure DevOps Pipelines Builds interface. At the top, there's a search bar and buttons for '+ New' and '+ Import'. Below it, a navigation bar has 'Mine' selected. The main area shows a list of build definitions: 'fabrikam build' and 'Fabrikam Fiber-CI'. For 'fabrikam build', a context menu is open, with the 'Add to my favorites' option highlighted with a red box. Other options in the menu include 'Queue new build...', 'Edit definition', 'Pause', 'View builds', 'Add to team favorites >', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', 'Security...', and '+ Add to dashboard >'.

Open **Build and Release>Builds** and choose either **Mine** or **Definitions** page. Choose the star icon next to the build definition you want to favorite. Or, open the context menu of the build definition, and then select **Add to**

my favorites or Add to team favorites.

The screenshot shows the 'Build Definitions' page in the Azure DevOps interface. At the top, there are tabs for 'Mine', 'All Definitions', 'Queued', and 'XAML'. A search bar at the top right contains the placeholder 'Build ID or build number' with a magnifying glass icon. Below the tabs, a table header includes columns for 'Recently built', 'Status', and 'Triggered by'. A single row is visible, representing a build definition named 'fabrikam build'. To the left of the definition name is a user icon with a checkmark and the text 'Recently built'. To the right are three small icons: a star, three dots, and a magnifying glass. The main content area shows a detailed view of the build definition, including its configuration and history. A context menu is open over the definition name, listing options such as 'Queue new build...', 'Edit...', 'View definition summary', 'Add to my favorites' (which is highlighted with a red box), 'Add to team favorites', 'Clone...', 'Export', 'Rename...', 'Save as a template...', 'Delete definition', and 'Security...'. The 'Add to my favorites' option is clearly distinguished by the red highlighting.

Favorite a test plan

To learn more about test plans, see [Create a test plan and test suite](#).

To mark a test plan as a favorite, open **Test Plans>Test Plans** and choose the star icon next to a test plan from the menu that shows All test plans.

To mark a test plan as a favorite, open the **Test>Test Plans** page and choose the star icon next to a test plan from the menu that shows All test plans.

Unfavorite a view you've favorited

You can unfavorite an artifact from your **Favorites** page. Choose the inbox icon, and then choose **Favorites**. Choose the favorited icon of a currently favorited artifact.

The screenshot shows the Microsoft Teams ribbon at the top. The tabs include Work Items, Pull requests, and Favorites, which is highlighted with a red box. Other tabs like Projects, Teams, Dashboards, Plans, and Queries are also visible. Below the ribbon, there's a list of favorite items.

Projects

Fabrikam Fiber

Teams

Phone

Voice

Web

Dashboards

Fabrikam Fiber Team Analytics

Plans

Backlog team plans

Fabrikam Fiber Feature plans

Queries

All items

All items on all projects

Assigned to me

Similarly, you can unfavorite an artifact from the same page where you favorited it.

You can unfavorite an artifact from the **Projects>Favorites** page and choose the favorited icon of a currently favorited artifact.

Similarly, you can unfavorite an artifact from the same page where you favorited it.

Try this next

[Follow a user story, bug, issue, or other work item or pull request](#)

Related articles

- [Manage personal notifications](#)
- [Set your preferences](#)

Work across projects

8/1/2019 • 4 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

When you work on several projects, you can use select pages to support your cross-project work. These pages provide quick access and filter functions to support your work within a single project or work you're doing across several projects.

For example, you can quickly access and navigate to the following artifacts defined across projects.

- **Projects:** Projects defined for an organization in Azure DevOps
- **Work items:** Work items assigned to you
- **Pull requests:** Pull requests you've initiated or that are relevant to you across all team projects you work in
- **Favorites:** Artifacts—such as projects, teams, repositories, shared queries, and more—that you've favorited
- **Projects:** Team projects and teams within the projects that you work in
- **Favorites:** Items—such as build definitions, repositories, shared queries, and more—that you've favorited
- **Work items:** Work items assigned to you, that you're following, or that you've recently viewed or updated
- **Pull requests:** Pull requests you've initiated or that are relevant to you across all team projects you work in
- **Projects:** Team projects and teams within the projects that you work in
- **Favorites:** Items—such as build definitions, repositories, shared queries, and more—that you've favorited
- **Work items:** Work items assigned to you, that you're following, or that you've recently viewed or updated
- **Pull requests:** Pull requests you've initiated or that are relevant to you across all team projects you work in
- **Rooms:** Team rooms you use to collaborate with other team members.

NOTE

The features described in this article require TFS 2017.1 and later versions. To upgrade to TFS 2017.1, go to the [Visual Studio downloads page](#).

To switch to another project, see [Switch project, repository, or team](#).

View and open work items

To view work items assigned to you across projects, choose the  inbox icon, and then choosing **Work Items**.

Work Items Pull requests Favorites

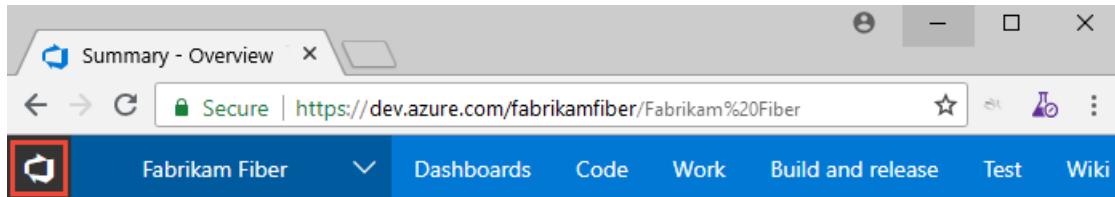
New query

Assigned to me

Design welcome screen Agile 11	● New
Check issues with permissions Fabrikam Fiber	● New
Research architecture changes Fabrikam Fiber	● New
Review security issues prior to development Agile 11	● Active
Welcome back page Fabrikam Fiber A	● New
Add an information form Fabrikam Fiber A	● New
Update welcome back page is slow Visual Studio Code	● New
XCode security test fails Visual Studio Code	● Active
Interim save on long forms Fabrikam Fiber	● New

To view work items assigned to you for a specific project, [open Work>Work Items>Assigned to Me](#).

Choose the Azure DevOps logo to open **Projects**.



Choose **Work items** to access the set of work items assigned to you or followed by you. The lists available from each page span all team projects that you work in.

The **Assigned to me** page lists all work items assigned to you in the order they were last updated. To open or update a work item, choose its title.

Projects Favorites Work items Pull requests ... | ⚙

My work items

Assigned to me Following My activity

Doing

397	Updates to the Web layout	● Resolved
200	Implement a factory which abstracts the email client	● In Progress
202	Integrate client app with IM clients	● In Progress
214	For/Not rules	● Proposed
198	Refactor compression code	● Proposed
288	Installer updates	● Proposed

Filter your work items...

Work you're following

Choose **Following** to open the page that lists all the work items [you've marked to follow](#). To stop following an item and remove it from your list, choose the following icon.

My work items

Assigned to me	Following	Mentioned	My activity
361	Interim save on long form	Christie Church	● Approved Fabrikam Fiber
384	Secure sign-in	Christie Church	● Committed Fabrikam Fiber
360	Change initial view	Raisa Pokrovskaya	● New Fabrikam Fiber

Projects Favorites Work items Pull requests ... | ⚙

My work items

Assigned to me **Following** My activity

140	Convert legacy OData service interfaces to Rest API	● In Progress
104	Create model report	● In Progress
580	Migrate legacy code to portable frameworks	● In Progress
621	Interim save on long form	● Completed
665	Check issues with permissions	● Resolved

My activity

Choose **My activity** to open the page that lists all work items that you have recently viewed or updated.

My work items

Assigned to me	Following	Mentioned	My activity			
386 Apply security measures	Johnnie McLeod	● To Do	Fabrikam Fiber	You viewed 6/1/2018		
516 Performance issues	Christie Church	● New	Fabrikam Fiber	You viewed 5/29/2018		
9 Story 1 - Do some work	Raisa Pokrovskaya	● In Progress	Fabrikam Fiber	You updated 5/22/2018		
435 I can add a task	Christie Church	● To Do	Fabrikam Fiber	You updated 5/22/2018		

Filter the list of work items

Similar to the Favorites page, you can filter the work pages by typing a keyword in the *Filter your work items...* box. The list will filter based on keyword matches to the work item ID, title, state, or team project name.

View and open pull requests

To view pull requests you created or that are assigned to you across projects, choose the inbox icon, and then choosing **Pull requests**.

The screenshot shows the 'Pull requests' section of the work items interface. At the top, there's a navigation bar with icons for search, filter (highlighted with a red box), and user profile. Below the bar, there are three tabs: 'Work Items', 'Pull requests' (which is highlighted with a red box), and 'Favorites'. The main content area is divided into two sections: 'Created by me' and 'Assigned to me'. Under 'Created by me', four pull requests are listed, each with a user icon, title, and commit details. Under 'Assigned to me', two pull requests are listed. All pull requests appear to be from the same user ('FabrikamFiber') and target the 'master' branch.

Created by me
Updated ReadMe.txt FabrikamFiber ↗ master
Updated FabrikamData.cs FabrikamFiber ↗ master
Updated Readme.txt FabrikamFiber ↗ users/jamal/international-address-support
Added support for international date formats FabrikamFiber ↗ master

Assigned to me
Updated Readme.txt FabrikamFiber ↗ users/jamal/international-address-support
Added date formats FabrikamFiber ↗ master

Open the **Projects>Pull requests** page to access any pull request that's relevant to you across all projects you work in. Choose **Active** or **Completed** to pivot between the active or completed set of pull requests. To initiate a pull request, choose **New pull request**.

My pull requests

Active Completed

Created by me



Updated README.md

Francis Totten requested #4 into master in Demo 2 minutes ago

Updated 3 hours ago



Assigned to me



Updated README.md

Raisa Pokrovskaya requested #2 into test in ginara 6 minutes ago

Updated 6/17/2016



From each page, you're one choice away from navigating to the branch or repository for a pull request. This mirrors capabilities on the project **Code>Pull Requests** page.

Filter the list of pull requests

Similar to the Favorites page, you can filter the list by typing a keyword in the *Filter pull requests* box.

View and open favorites

You can view favorites you've set across projects. To learn more about working with favorites, see [Set personal or team favorites](#).

Choose the inbox icon, and then choosing **Favorites**.

The screenshot shows the 'Favorites' page in Azure DevOps. At the top, there are navigation links for 'Work Items' and 'Pull requests'. Below them is a red-bordered box around the 'Favorites' link. To the right are icons for search, filter, and user profile. The main content area is divided into sections: 'Projects', 'Teams', 'Dashboards', 'Plans', and 'Queries'. Each section lists items with a star icon indicating favoritism. The 'Favorites' link is underlined.

Section	Item	Favorites
Projects	Fabrikam Fiber	★
	Phone	★
	Voice	★
Teams	Web	★
	Fabrikam Fiber Team Analytics	★
	Backlog team plans	★
Plans	Fabrikam Fiber Feature plans	★
	All items	★
	All items on all projects	★
Queries	Assigned to me	★

Open the **Favorites** page to quickly access any object or item that you've marked as a favorite.

The screenshot shows the 'Favorites' page with a 'Filter favorites' search bar at the top right. The main content area is titled 'Queries' and lists five items, each with a star icon indicating favoritism. The items are: 'Bug Triage' (Fabrikam Fiber, Shared Queries/Current Iteration), 'My Bugs' (Contoso, Shared Queries), 'Open User Stories' (Contoso, Shared Queries/Current Iteration), 'Product Planning' (Fabrikam Fiber, Shared Queries), and 'Product Planning' (Contoso, Shared Queries).

Query Type	Owner	Location	Favorites
Bug Triage	Fabrikam Fiber	.../Shared Queries/Current Iteration	★
My Bugs	Contoso	Shared Queries	★
Open User Stories	Contoso	.../Shared Queries/Current Iteration	★
Product Planning	Fabrikam Fiber	Shared Queries	★
Product Planning	Contoso	Shared Queries	★

To learn more about working with favorites, see [Set personal and team favorites](#).

Favorite and unfavorite an item

To mark any item as a favorite, choose the ★ star icon next to the object.

To remove an item from your favorites list, choose the favorited icon.



Filter the list of favorites

To filter the list, type a keyword in the *Filter favorites* box. The list will filter based on keyword matches to the title or team project name associated with the favorited item.

Open a team room

To open a team room, choose **Projects>Rooms**. You'll see all the team rooms defined for collection of projects. Choose the name of a team room which you have access. You only have access to those team rooms of which you are a team member.

The screenshot shows the TFS Team Room interface. At the top, there's a navigation bar with 'fabrikam' logo, 'Home', 'Users', 'Rooms', 'Load test', and a gear icon. On the left, a sidebar lists 'Contoso Team Room', 'Design Room', 'Engineering Room', 'Fabrikam Fiber Team Room' (which is selected and highlighted in blue), and 'Research Room'. The main area is titled 'Fabrikam Fiber Team Room' with a date '10/24/2016' and a 'Live' status. A message from 'Bug #462' says 'Avatar does not load in home page.' A reply from '...Pokrovskaya' asks if it's the bug mentioned. 'Jamal Hartnett' responds that it is and offers to take a look. '...Pokrovskaya' agrees to do so. 'Jamal Hartnett' then thanks 'Johnnie McLeod' for a changeset update. Below this, a build summary for 'AL.Gated 1302.144' shows a failure. At the bottom, there's a 'Post a message' input field.

To learn more about team rooms, see [Collaborate in a team room](#).

NOTE

Team Rooms are no longer supported for TFS 2018 and later versions as described in [Deprecation of Team Rooms](#) blog post. Several good solutions are available that integrate well with TFS that support notifications and chat, such as [Microsoft Teams](#) and [Slack](#).

Related articles

- [Enable preview features](#)
- [Connect to team projects](#)

Filter lists, boards, and directories

Filter lists and boards

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Several applications and pages support filtering, which is very useful when a large number of artifacts or items have been defined. Most directory views provide one or more filter functions.

You can filter most items using keywords or a user name for an author of an item or where work is assigned to them. You can filter lists and boards in the following areas:

- Git repositories: Branches, Commits, Commit history, Pull Requests, Pushes, and Repositories
- Work tracking: Work Items, Kanban boards, Backlogs, Sprint Backlogs, and Taskboards
- Directories: Dashboards, Boards, Backlogs, Sprints, Queries, Builds, Releases
- Git repositories: Branches, Commits, Commit history, Pull Requests, Pushes, and Repositories
- Work tracking: Work Items, Kanban boards, Backlogs, Sprint Backlogs, and Taskboards

NOTE

You may have fewer or additional filter options based on the [features you've enabled](#) or the platform and version that you are working from.

Filter based on keywords, tags, or fields

To turn filtering on, choose the  filter icon.

You can filter work items by typing a keyword or using one or more of the fields provided, such as work item type, assigned to, state, and tags. Based on the keyword that you enter, the filter function will list work items based on any visible/displayed column or field, including tags. Also, you can enter a value for an ID, whether or not the ID field is visible.

Backlog items

Backlog Board Forecast Off Parents Hide In progress items Show Mapping Off  

New |   Create query | Column options | 

 Filter by keyword Types Assigned to

States Tags  Clear

Type Product Backlog Item  Add

Order	State	Title	Tags	Iteration Path
1	Approved	Hello World Web Site	...	Fabrikam Fiber\Release 1\Sprint 11
2	New	Change initial view	Web	Fabrikam Fiber\Release 1\Sprint 9
3	New	Slow response on information form		Fabrikam Fiber\Release 1\Sprint 9

The filtered set is always a flat list, even if you've selected to show parents.

Characters ignored by keyword filter criteria

The filter criteria ignores the following characters: (comma), (period), (forward slash), and (backslash).

Filter work items based on keywords

You can use keywords to filter your backlogs or queries. The filter function lists those work items based on any visible/displayed column or field, including tags, based on the keyword that you enter. Also, you can enter a value for an ID, whether or not the ID field is visible.

Here, we filter the backlog to only show items that include 'Web' in any one of the displayed column fields.

Backlog items



Backlog			Board	Parents	Show	In progress items	Show	Mapping	On	⚙️	[]
New	[+]	[]	Create query	Column options	[]	[]	[]	[]	[]	[]	[]
State	Effort	Title	Tags								
● New	...	👑 Improve User Experience	Web								
● New		🏆 Scale user interface appropriate to device	Web								
● Committed	8	🐞 Slow response on welcome page	Web								
● Committed	5	🐞 Secure sign-in	Mobile Web								
● Committed	8	📘 Interim save on long forms	Web								
● New		📘 Hello World Web Site									
● New	5	📘 Change the initial view	Web								
● New		📘 Add animated emoticons	Web								
● New	3	📘 Welcome back page	Web								
● New		📘 Log on	Web								
● New		📘 Research architecture stories	Web								
● New		📘 Search on web									

The filtered set is always a flat list, even if you've selected to show parents.

Characters ignored by keyword filter criteria

The filter criteria ignores the following characters when the field value starts with the character:

{, (, [, !, @, #, \$, %, ^, &, *, ~, ` , ', ".

Filter work tracking backlogs and queries based on tags

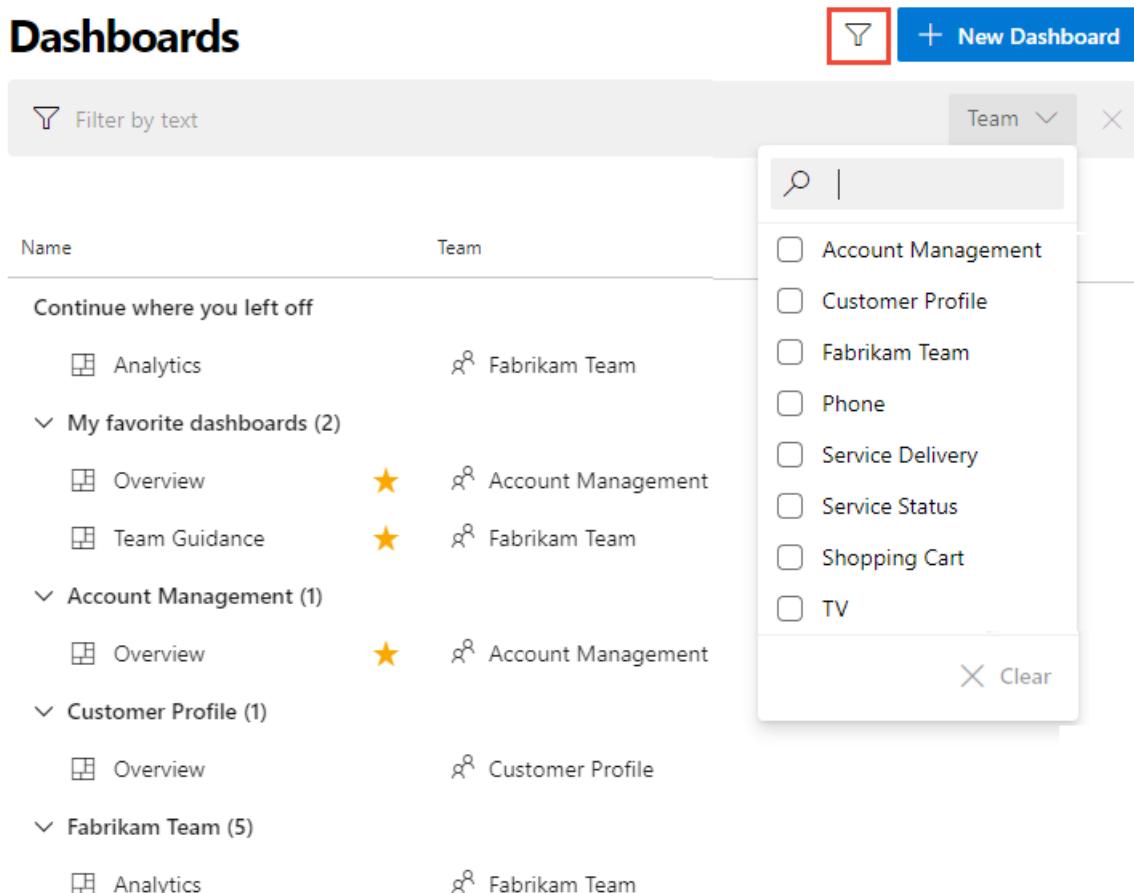
If you've [added tags to your work items](#), you can filter your backlogs, Kanban boards, and query results using the  tag filter. For backlogs and query results, add Tags as a column option prior to filtering on tags.

To learn more about filtering using tags, see [Add tags to work items to categorize and filter lists and boards](#), [Filter lists using tags](#)

Filter directories

Choose the  filter icon to filter a directory list by keyword, team, or other supported field. Keywords apply to titles, descriptions, and team names.

For example, here we turn filtering on for the dashboard directory.



The screenshot shows a 'Dashboards' list page. At the top right, there is a filter icon () and a '+ New Dashboard' button. Below the header, there is a search bar labeled 'Filter by text' with a magnifying glass icon. A dropdown menu is open under the 'Team' label, showing a list of teams with checkboxes: Account Management, Customer Profile, Fabrikam Team, Phone, Service Delivery, Service Status, Shopping Cart, and TV. A 'Clear' button is at the bottom of the dropdown. The main list table has columns for 'Name' and 'Team'. The data in the table is as follows:

Name	Team
Continue where you left off	
Analytics	Fabrikam Team
My favorite dashboards (2)	
Overview	Account Management
Team Guidance	Fabrikam Team
Account Management (1)	
Overview	Account Management
Customer Profile (1)	
Overview	Customer Profile
Fabrikam Team (5)	
Analytics	Fabrikam Team

Related articles

- [Commit history](#)
- [Working with Git tags](#)
- [Filter backlogs and queries](#)
- [Filter your Kanban board](#)
- [Add tags to work items](#)

Search across your code base, work items, or wiki

8/1/2019 • 6 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

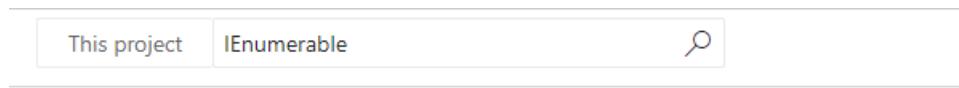
With the search box, you can quickly find a code file, work item, or wiki page.

With the search box, you can quickly find a code file or work item.

Initiate a code search

Code search requires installation of the [Code Search extension](#). If it hasn't yet been installed, request that a member of your Project Collection Administrators group [install it](#).

To start your search, choose **Repos>Files** or other page under **Code**, enter a keyword or phrase in the search box, and press *Enter* or choose the  start search icon.



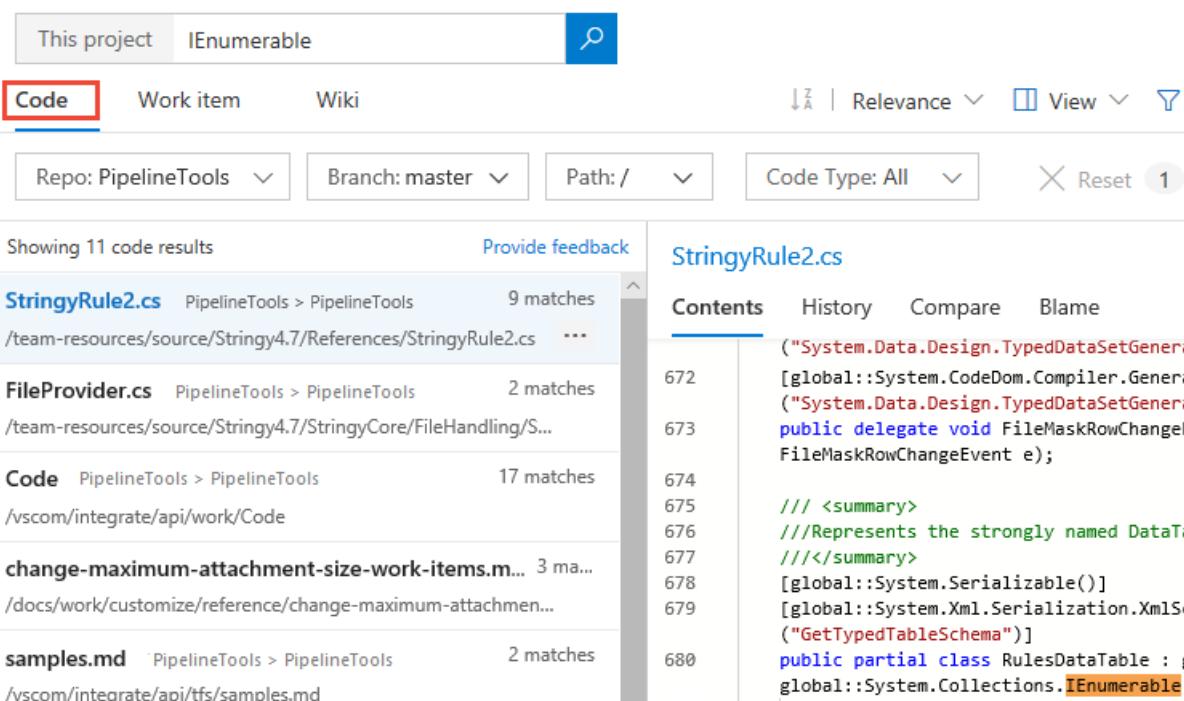
1. In the search box, check that the text displays *Search code*. If it doesn't, use the selector to select it.



2. Enter a search string in the text box, and press *Enter* (or choose the  icon) to start your search.

Work with code search results

1. The search page shows a list of the matching code files. The selected file has all instances of the search string highlighted. If you see a list of work items, ensure that **Code** is selected in the top left.



This project IEnumerable 

Code Work item Wiki  Relevance  View 

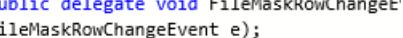
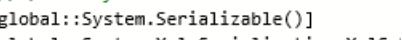
Repo: PipelineTools  Branch: master  Path: /  Code Type: All  Reset 1

Showing 11 code results [Provide feedback](#)

StringyRule2.cs	PipelineTools > PipelineTools	9 matches
/team-resources/source/Stringy4.7/References/StringyRule2.cs ...		
FileProvider.cs	PipelineTools > PipelineTools	2 matches
/team-resources/source/Stringy4.7/StringyCore/FileHandling/S...		
Code	PipelineTools > PipelineTools	17 matches
/vscom/integrate/api/work/Code		
change-maximum-attachment-size-work-items.m... 3 ma...		
/docs/work/customize/reference/change-maximum-attachmen...		
samples.md	PipelineTools > PipelineTools	2 matches
/vscom/integrate/api/tfs/samples.md		

StringyRule2.cs

Contents History Compare Blame

672 
673 
674 
675 
676 
677 
678 
679 
680 

2. Try assembling more complex search strings using the operators and functions listed in the handy drop-

down list. Select the filter function or code type you want to include in your search string from the list, and type the criteria value.

Filter by scope (e.g., Activity ext:cs)

ext: With file extension
file: Filename
path: Under path
proj: Inside project
repo: Inside repository

Filter by code type (e.g., funcApiRoot)

arg: Argument
basetype: Basetype
caller: Caller
class: Class
classdecl: Class Declaration
classdef: Class Definition
comment: Comment
ctor: Constructor
decl: Declaration
def: Definition
▼ Show more

Operators Operators Example
AND NOT OR

For more ways to search, see the [help page](#)

- You can find all instances of "ToDo" comments in your code simply by selecting `comment: todo`.
- You can search in specific locations, such as within a particular path, by using a search string such as `Driver path:MyShuttle/Server`.
- You can search for files by name, such as `Driver file:GreenCabs.cs`, or just by file extension. For example, the search string `error ext:resx` could be useful when you want to review all error strings in your code. But even if your plain text search string (without specific file type functions) matches part of a filename, the file appears in the list of found files.
- You can combine two or more words by using Boolean operators; for example, `validate OR release`.
- You can find an exact match to a set of words by enclosing your search terms in double-quotes. For example, `"Client not found"`.
- You can use the code type search functions with files written in C#, C, C++, Java, and Visual Basic.NET.

To learn more, see [Search code](#).

Initiate a work item search

1. Choose any **Boards** page, enter a keyword or phrase in the search box, and press *Enter* or choose the start search icon.

2. Search results are displayed in a snippet view where the matches found are shown in bold.

This screenshot shows the Azure DevOps search interface. The search bar at the top contains the text "login". Below the search bar, there are filters for "Area: FabrikamFiber Web" and "Types: All". The results section displays three work items:

- 62 Login page**: Resolved. Description: Login page.
- 119 Login behaviour for booking**: Closed. Description: Login behaviour for booking.
- 97 Login and logout behaviours**: Closed. Description: Login and logout behaviours.

The right pane provides detailed information for the selected work item, "62 Login page". It includes fields for State (Resolved), Area (FabrikamFiber W), Reason (Code complet ...), and Iteration (FabrikamFiber W).

This is a full text search that uses simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Note that searches are *not* case-sensitive.

3. Select a snippet of a work item to display it in the right window.

Open the search results in a new browser tab from a search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the icon. In Google Chrome, press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

1. In the search box, check that the text says *Search work items*. If it doesn't, use the selector to select it.

A screenshot of a search bar. The placeholder text "Search work items" is visible, along with a magnifying glass icon and other icons for filter, save, and user profile.

2. Enter a search string in the text box, and press *Enter* (or choose the icon) to start your search.

3. Search results are displayed in a snippet view where the matches found are shown in bold.

This screenshot shows the Azure DevOps search interface again, with the search bar containing the text "login". The results section displays the same three work items as the previous screenshot. The right pane shows the details for the "62 Login page" work item.

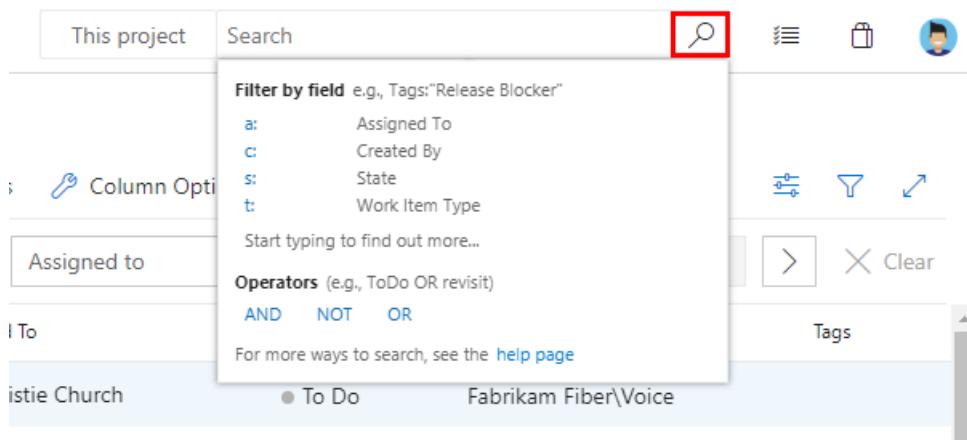
This is a full text search that uses simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Note that searches are *not* case-sensitive.

4. Select a snippet of a work item to display it in the right window.

Open the search results in a new browser tab from a search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the  icon. In Google Chrome, press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

Fine tune your work item search results

1. Fine tune your search by specifying the fields to search. Enter and a user name to search for all items assigned to that user.

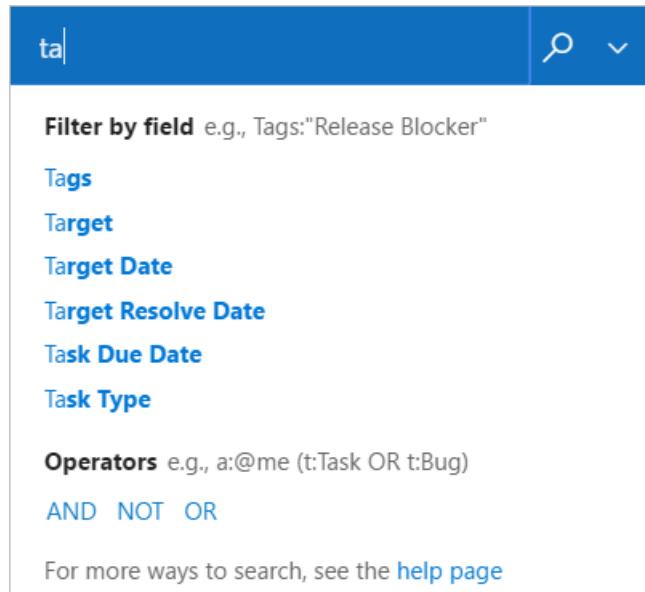


The screenshot shows the search interface in the Azure DevOps portal. A red box highlights the search bar at the top. Below it, a dropdown menu titled "Filter by field" lists quick filters: "Assigned to", "Created by", "State", and "Work Item Type". The "Assigned to" filter is selected. The search bar contains "a: istie Church". At the bottom, there are buttons for "To Do" and "Fabrikam Fiber\Voice".

The quick filters you can use are:

- for **Assigned to**:
- for **Created by**:
- for **State**
- for **Work item type**

2. Start typing the name of a field in your work items; for example, type .



The screenshot shows the search interface with the input "ta" in the search bar. A dropdown menu titled "Filter by field" lists suggestions: "Tags", "Target", "Target Date", "Target Resolve Date", "Task Due Date", "Task Type", and "Operators". The "Tags" suggestion is highlighted. The search bar also contains "ta". At the bottom, there is a link "For more ways to search, see the help page".

The dropdown list shows work item field name suggestions that match user input thereby helping the user to complete the search faster. For example, a search such as **tags:Critical** finds all work items tagged 'Critical'.

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if required. For example, **a: Chris t: Bug s: Active** finds all active bugs assigned to a user named Chris.
4. Narrow your search to specific types and states, by using the drop-down selector lists at the top of the

results page.

1. Fine tune your search by specifying the fields to search. Enter a: and a user name to search for all items assigned to that user.

The screenshot shows the search interface with a sidebar containing quick filters:

- Filter by field** e.g., Tags:"Release Blocker"
- a:** Assigned To
- c:** Created By
- s:** State
- t:** Work Item Type

Below the filters is a text input field: "Start typing to find out more..." and a section for **Operators** (e.g., ToDo OR revisit) with options AND, NOT, OR. At the bottom, it says "For more ways to search, see the [help page](#)".

The quick filters you can use are:

- a: for **Assigned to**:
- c: for **Created by**:
- s: for **State**
- t: for **Work item type**

2. Start typing the name of a field in your work items; for example, type ta.

The screenshot shows the search interface with the input field containing "ta". Below it is a dropdown list of work item field name suggestions:

- Tags
- Target
- Target Date
- Target Resolve Date
- Task Due Date
- Task Type

Below the suggestions is a section for **Operators** e.g., a:@me (t:Task OR t:Bug) with options AND, NOT, OR. At the bottom, it says "For more ways to search, see the [help page](#)".

The dropdown list shows work item field name suggestions that match user input thereby helping the user to complete the search faster. For example, a search such as **tags:Critical** finds all work items tagged 'Critical'.

3. Add more filters to further narrow your search, and use Boolean operators to combine terms if required. For example, **a: Chris t: Bug s: Active** finds all active bugs assigned to a user named Chris.
4. Narrow your search to specific types and states, by using the drop-down selector lists at the top of the results page.

To learn more, see [Search work items](#)

Initiate a wiki content search

With wiki search, you can search within a project wiki or across all wikis created for the organization. Simply select

the **Search wiki** option from the search box and type a keyword or phrase within the search box.

NOTE

Wiki search is supported on TFS 2018.2 or later versions. To download TFS 2018.2, see [Team Foundation Server 2018 Update 2 Release Notes](#).

Choose **Overview>Wiki** and type your keyword or phrase into the search box.

The screenshot shows the Azure DevOps interface for a project named 'Fabrikam Fiber'. The top navigation bar has tabs for 'This project' and 'Request', with 'Request' being the active tab and highlighted with a red box. The left sidebar contains links for Overview, Summary, Dashboards, Analytics views, Wiki (which is also highlighted with a red box), Boards, and Repos. The main content area is a 'Wiki' page titled 'SDK repo > README'. It features sections like 'Introduction' and 'Getting Started', each with a 'TODO' note. At the bottom of the page, there's a link to 'Installation process'.

The search feature quickly returns wiki pages by title or page content. English language stemming support helps you find the most relevant wiki pages. For example, when you enter *request* in the search box, wiki search will return page results containing related words such as *requesting*, *requested*, *requests*, and so on.

The screenshot shows the search results for 'Request' in the 'Wiki' section of the Azure DevOps interface. The search bar at the top shows 'Request' selected. Below it, a dropdown menu shows 'Wiki: All Wikis'. The search results list several entries:

- Request extensions**
in Fabrikam Fiber > Fabrikam Fiber.wiki
- slack**
in Fabrikam Fiber > Fabrikam Fiber.wiki
For example, the pull request created trigger can be filtered on the repository in which the pull request occurs., the target branch it applies to, and the team members that are required or invited to review the request. Q & A. ####...
- follow work items**
in Fabrikam Fiber > Fabrikam Fiber.wiki
The Follow a pull request feature is available from Team Services and TFS 2017 Update 1. ##Follow a work item or pull request. When you want to track the progress of a single work item, click the icon. ##Follow a pull request. To...
- overview**
in Fabrikam Fiber > Fabrikam Fiber.wiki
/connect/account-home-pages.md). [Pull requests relevant to me](.. The first is through the Request feedback link that interacts with the Microsoft feedback client. Microsoft Feedback client. [Request feedback](..

Open **Wiki** and type your keyword or phrase into the search box.



The search feature quickly returns wiki pages by title or page content. English language stemming support helps you find the most relevant wiki pages. For example, when you enter *request* in the search box, wiki search will return page results containing related words such as *requesting*, *requested*, *requests*, and so on.

This screenshot shows the Microsoft Teams search interface. At the top, there are tabs for 'This project', 'Request', and a magnifying glass icon. Below the tabs, there are links for 'Code 0', 'Work item 1', and 'Wiki 29'. A search bar contains the text 'Wiki: All Wikis' with a dropdown arrow and a 'Clear' button. To the right of the search bar is a link to 'Search this organization' with a blue arrow icon. Below the search bar, it says 'Showing 29 results'. The results list includes:

- Request extensions**
in Fabrikam Fiber > Fabrikam Fiber.wiki
- slack**
in Fabrikam Fiber > Fabrikam Fiber.wiki
For example, the pull request created trigger can be filtered on the repository in which the pull request occurs., the target branch it applies to, and the team members that are required or invited to review the request. Q & A. ####...
- follow work items**
in Fabrikam Fiber > Fabrikam Fiber.wiki
The Follow a pull request feature is available from Team Services and TFS 2017 Update 1. ##Follow a work item or pull request. When you want to track the progress of a single work item, click the icon. ##Follow a pull request. To...
- overview**
in Fabrikam Fiber > Fabrikam Fiber.wiki
/connect/account-home-pages.md). [Pull requests relevant to me](.. The first is through the Request feedback link that interacts with the Microsoft feedback client. Microsoft Feedback client. [Request feedback](..

Related articles

- [Search code](#)
- [Search work items](#)
- [Create a wiki for your project](#)

Enable preview features

9/24/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

As new features are introduced, you can turn them on or off. That way, you can try them out, provide feedback, and work with those features that meet your requirements.

Some features provide a new user interface and functionality, which can be managed per user or team member. Others support a default experience for the account and are managed by an account administrator.

NOTE

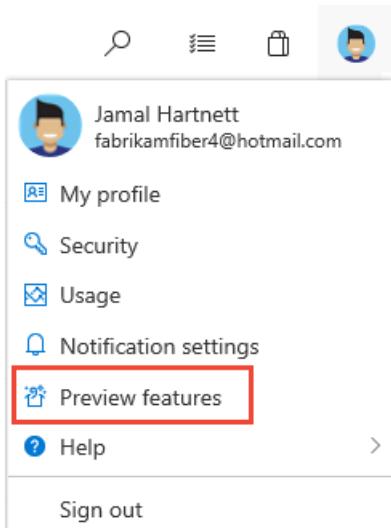
You can turn on or off select features for Azure DevOps Services. Preview features become available first on Azure DevOps Services and then become standard features with an update to Azure DevOps Server. At some point, the preview feature moves out of preview status and becomes a regular feature of the web portal.

PREVIEW FEATURES PER USER	PREVIEW FEATURES PER ORGANIZATION
<ul style="list-style-type: none">- Analytics Views- Enable modern search- Experimental themes- Multi-stage pipelines- New Boards reports- New release progress views- New service connections experience- New Test Plans Page- New Test Plans Page- New user hub- New Wiki experience- Organization Permissions Settings Page v2- Project Permissions Settings page- Test tab in new web platform	<ul style="list-style-type: none">- Analytics Views- Experimental themes- Full Access to Azure Pipelines for Stakeholders- Git Forks- Multi-stage pipelines- New Boards reports- New release progress views- New service connections experience- New Test Plans Page- New user hub- New Wiki experience- Organization Permissions Settings Page v2- Pipeline decorators- Project Permissions Settings page- Test tab in new web platform

Enable features for your use

From time to time, a new feature is introduced in Preview mode, which allows you to turn it on or off.

1. To access the Preview features options, open your profile menu, and select **Preview features**.



2. To enable or disable a feature, choose the slider.

X

Preview features

The following preview features are available for your evaluation. Help us make them better!

for me [Jamal Hartnett] ▾

Experimental Themes On
Adds an early preview of various themes to the Theme management panel.

Git commit menu extension points On
Opens up Git commit context menu to contributed actions.

New build result page On
Lights up new build results page.

New log reader for Pipelines On
Lights up new logs reader for Pipelines.

New PAT Experience On
Enable new Personal access token page on security hub.

New release progress views On
Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

Enable features at the organization level (for all users)

When you enable a feature at the organization level, you essentially turn it on for all users of your account. Each user can then disable the feature if they so choose.

TIP

If you don't see the **for this account** menu option, then you aren't an account administrator. To get added as one, see [Add administrators, set permissions at the team project or collection level](#).

Preview features

X

The following preview features are available for your evaluation. Help us make them better!

for this account [fabrikam]

Experimental Themes

On

Adds an early preview of various themes to the Theme management panel.

Full access to Azure Pipelines for Stakeholders

On

Gives users with the Stakeholder license full access to Azure Pipelines for private projects. Limit what they can do by using security groups and permissions. Turning on this feature doesn't affect public projects, where Stakeholders always have full access. [Learn more](#)

Git Forks

On

Enable git repositories to be forked. [Learn more](#)

New build result page

On

Lights up new build results page.

New log reader for Pipelines

Off

Lights up new logs reader for Pipelines.

New PAT Experience

On

Enable new Personal access token page on security hub.

New release progress views

On

Turn on the new release views to visualize the progress of your deployment pipelines. [Learn more](#)

New Releases Hub

On

Turns on the experience to create folders and manage release pipelines.

New YAML pipeline creation experience

Off

Enables the new YAML pipeline creation experience.

Test analytics in new web platform

On

Lights up test analytics features in new web platform.

Test tab in new web platform

On

Lights up a new test tab under build in new web platform.

Features now enabled for all Azure DevOps Services

General

- New PAT experience
- New Navigation

Azure Pipelines

- Test analytics in new web platform
- New builds hub
- Build with multiple queues
- New Releases Hub
- Approval gates in releases - New Release Definition Editor
- Symbol server
- Task tool installers

Azure Boards

- New Rich Text Editor- New Queries Experience
- New Work Items

Azure Repos

- Pull Request Status Policy

Azure Artifacts

- NuGet.org upstream sources
- Updated package experience

Azure Test Plans

- New Test Plan Experience

Dashboards and Analytics

- New Dashboards Experience

Social tools

- Wiki
- Combine email recipients
- New experience in Code, Work Item, & Wiki search
- Out of the box notifications
- Team expansion for notifications

Organization, project, and billing management

- Streamlined User Management

Project management and navigation glossary

8/1/2019 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

This glossary describes terms used when navigating in the web portal for Azure DevOps. See also:

- [Agile glossary](#)
- [Security glossary](#)

Backlogs

An interactive list of work items that corresponds to a team's project plan or roadmap for what the team plans to deliver. The product backlog supports prioritizing work, forecasting work by sprints, and quickly linking work to portfolio backlog items. You can define your backlog items and then manage their status using the Kanban board.

Each product backlog can be customized by a team. Learn more: [Create your backlog](#).

Analytics views

Analytics views provide a simplified way to specify the filter criteria for a Power BI report based on the Analytics service. The Analytics service is the reporting platform for Azure DevOps Services.

Area path

Area paths are used to group work items by team, product, or feature area. Iteration paths are used to group work into sprints, milestones, or other event-specific or time-related periods. You can use area paths to define a hierarchy of paths. To learn more, see [About area and iteration paths](#).

Boards (Kanban)

An interactive, electronic sign board that supports visualization of the flow of work from concept to completion and lean methods. Learn more: [Kanban basics](#).

Collections

A collection is a container for a number of projects in Azure DevOps. A default collection is created when you sign up with Azure DevOps Services or install Team Foundation Server. Within Azure DevOps Services, a collection corresponds to an organization. For on-premises TFS deployments, you can add and manage collections to specify the logical and physical resources available to the projects within the collection.

Learn more: [About projects and scaling your organization](#), [Manage organizations](#) or [Manage project collections in Team Foundation Server](#).

Dashboards

Dashboards are user-configurable interactive signboards that provide real-time information. Dashboards are associated with a team and display configurable widgets to show information. To learn more, see [Add and manage dashboards](#).

Extensions

Extensions are simple add-ons that are used to customize and extend the DevOps experience of Azure DevOps Services and TFS. They are written with standard technologies—HTML, JavaScript, CSS—and can be developed using your preferred development tools. Hundreds of extensions are available from the [Visual Studio Marketplace](#), Azure DevOps tab.

Favorites

Tagging an object as a favorite is a method used to support quick navigation by yourself or other team members. You can tag work item queries and build definitions as personal and team favorites. Other objects you can tag as a favorite for yourself only include code branches, delivery plans, test plans, and teams or projects. To learn more, see [Set personal or team favorites](#).

Follow

Tagging specific work items or pull requests to follow them is a method used to receive email updates about changes that are made to them. To learn more, see [Follow a work item or pull request](#).

Git repository

A Git repository supports a distributed version control system for tracking changes, reviewing contributions to the code, and more. Each developer has a copy of the source repository on their dev machine. You can add multiple Git repositories to a project. Learn more: [Git Repositories](#).

NOTE

Git in Visual Studio and Azure DevOps Services is standard Git. You can use Visual Studio with third-party Git services, and you can also use third-party Git clients with Azure DevOps Services.

Notifications

With notifications, you receive an email when changes occur to work items, code reviews, pull requests, source control files, and builds. For example, you can get notified whenever a bug that you opened is resolved, or when a work item is assigned to you. You receive notifications based on rules or subscriptions made by you, for your teams, or for the project. Learn more: [About notifications](#).

Pipelines

Pipelines are artifacts that you define to run concurrent builds or deploy concurrent releases. Two types of pipelines are supported, private and hosted. To learn more, see [CI/CD concurrent jobs](#).

Plans (also known as delivery plans)

A plan is a configurable view that displays work from multiple teams and projects laid out within a calendar based on each team's iterations. Each row in the view represents the work from a team's product or portfolio backlog. Each card corresponds to a work item, such as user story, feature, or epic. To learn more, see [Review team delivery plans](#).

Process

A process defines the building blocks of a work-tracking system. To customize a process, you first create an inherited process from one of the default system processes, [Agile](#), [Scrum](#), or [CMMI](#). All projects that use the

process see the changes you make. To learn more, see [About process customization and inherited processes](#).

Projects

A project, which was previously known as a *team project*, provides a repository for source code. A project provides a place where a group of people can plan, track progress, and collaborate on building software solutions. A project is defined for an Azure DevOps Services organization or within a TFS project collection. You can use it to focus on those objects defined within the project. To learn more, see [About projects and scaling your organization](#).

Public projects

A project created within an Azure DevOps Services organization that is visible to the whole world. Everyone in the world can discover them and perform limited operations. Administrators can control who gets to fully contribute. Administrators can switch a project from private to public, and vice-versa, as described in [Change the project visibility](#).

Queries

Queries are used to find and list work items. Queries support managed searches, which are used to triage work, versus ad-hoc searches, which are used to find a specific work item. Flat-list queries also support status and trend charts. To learn more, see [About managed queries](#).

Repositories

A source control folder or container you configure to help you track file changes in. You can have any number of repository on your computer, each stored in their own folder. Each repository is independent, so changes saved in one repository don't affect the contents of another. Learn more: [Create a new Git repo](#).

Sprints (also known as iterations)

A sprint is a time period of usually two to three weeks that's used to group work items to be completed during that time period. Sprints are used in Scrum methods to support sprint planning, sprint burndown, and other Scrum processes. Sprints are defined via iteration paths. To learn more, see [About area and iteration paths \(aka sprints\)](#).

Sprint backlog

An interactive list of work items that have been assigned to the same sprint or iteration path for a team. The sprint backlog supports teams that use Scrum methodologies. Learn more: [Sprint planning](#).

Taskboard

A taskboard is an interactive board of work items that you can use to review and update tasks defined for the sprint backlog. The taskboard supports teams that use Scrum methodologies. To learn more, see [Update and monitor your taskboard](#).

Teams

A team corresponds to a selected set of project members. With teams, organizations can subcategorize work to better focus on all the work they track within a project. Each team gets access to a suite of Agile tools. Teams can use these tools to work autonomously and collaborate with other teams across the enterprise. Each team can configure and customize each tool to meet their work requirements. To learn more, see [About teams and Agile tools](#).

Team Foundation Version Control (TFVC)

A centralized version control system. With TFVC, devs have only one version of each file on their dev machines. Branches are path-based and created on the server. Historical data is maintained only on the server. Learn more: [Use Team Foundation Version Control](#).

Widgets

Widgets display information and charts on dashboards. Many of them can be configured. Many widgets display information available from one or more data stores or charts created by the system. To learn more, see [Widget catalog](#).

Work items

A work item represents an object stored in the work item data store. Each work item is based on a work item type—such as a user story, feature, bug, task, or issue—and is assigned an identifier which is unique across all projects in an organization or project collection. The work item types available to you are based on the process used when your project was created. Each work item supports capturing information, adding attachments, linking to other work items, and more. Learn more: [About work items](#).

Work item types (WITs)

A WIT specifies the fields, workflow, and form used to track an item of work. Each WIT is associated with more than 30 system fields and several more type-specific fields. You use work items to plan and track the work required to develop your project. For an overview of predefined WITs provided with the default processes, see [Choose a process](#).

View and update work items via the mobile browser

9/3/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018

With the mobile browser and work item form, you gain on-the-go features to stay on top of the latest updates made to work tracking. When you click any work item link on your mobile device, it will open a mobile-friendly version of the work item. From there, you can update the work item or access all work items assigned to you or that you're following.

The screenshot shows a mobile browser displaying a work item form for a 'USER STORY 1'. The header bar is blue with the title 'Work items' and a message 'Mobile form is in PREVIEW.' A close button 'X' is in the top right corner. Below the header, the work item title 'USER STORY 1' is shown with a small icon. The main content area contains the text 'This is the new mobile experience!' followed by the last update information: 'Updated Wednesday by Jamal Hartnett'. Below this, a comment from 'Jamal Hartnett' is displayed, reading '@Chuck Reinhart - Note the differences between the previous and the latest version.' A 'View 1 comment >' link is present. At the bottom of the main content area, there are tabs for 'Details', 'History', 'Links', and 'Attachments', with 'Details' being the active tab. The 'Assigned To' section shows 'Christie Church' assigned to the story. The 'State' section shows 'New' with a 'Reason' of 'New'. The 'Area' and 'Iteration' sections both show 'Fabrikam Fiber'. The 'Description' section has a collapse/expand arrow '^' at the bottom right. The footer of the browser window shows navigation icons for back, forward, search, and refresh.

NOTE

The mobile browser supports Azure DevOps work tracking. To sign up for free, go to [Azure DevOps Services](#). The mobile browser is not an app, but a mobile view into select features. There is nothing to download. You access the mobile browser by clicking a link from a work item you receive in your mobile email application.

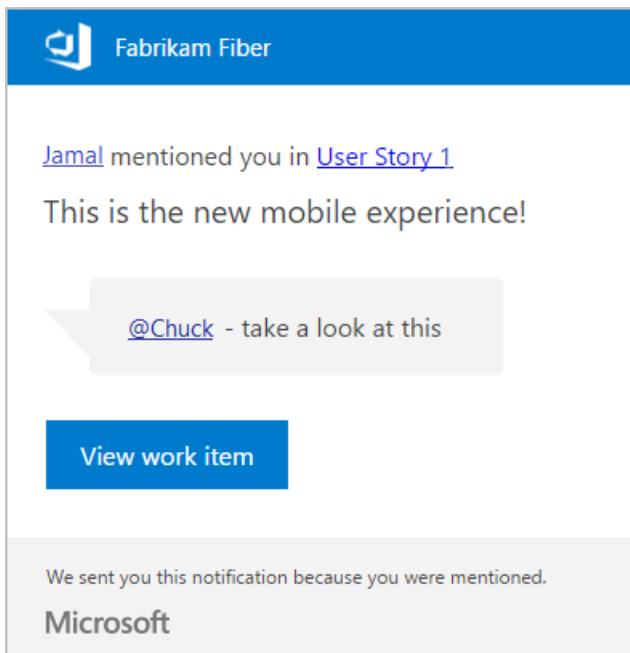
NOTE

The mobile browser is available for Azure DevOps Server 2019 and TFS 2018. For downloads, see [Downloads](#). The mobile browser is not an app, but a mobile view into select features. There is nothing to download. You access the mobile browser by clicking a link from a work item you receive in your mobile email application.

Open the mobile work item form

The mobile work item form will open when you click **View work item** from an email you receive from your mobile device. You'll receive this type of email under these circumstances:

- Changes were made to a work item you're following
- You were **@mentioned** in a discussion
- A notification is sent based on the work item alerts you've set using [Manage personal notifications](#).



Update a work item

Within the mobile form, you can do almost everything you can do from the [web portal form](#). Here are the actions you can take in the order they appear in the mobile form:

- Add and remove tags
- View and add to the discussion, click on the comment to add to the discussion
- View and update any field within the form (Assign to, State, Area, Iteration, Description, and more)
- View and open a link within the Development section
- View History
- View and open a link from the Links tab
- Open and add an attachment from the Attachments tab

Actions not available to you within the mobile work item form:

- You can't create or add new work items
- You can't initiate a development operation
- You can't add a link

Interact with mobile form controls

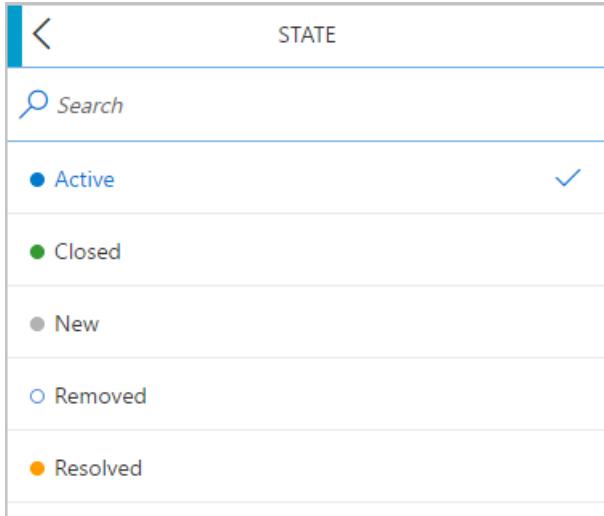
Mobile form controls operate as follows:

- Click any field to edit it and the form changes to a full-screen experience. For example, some of the most common actions such as changing the state of an item, moving to a different area path, adding an attachment, and creating/removing tags are all supported.

- When done, click the  return option.
- Remember to click the  save icon to save your changes!

Update status (change State)

To update the state, click the state you want.



STATE	Selected
Active	✓
Closed	
New	
Removed	
Resolved	

Add or remove tags

To add a tag, type the text you want.



TAGS
+ Add tag
mobile
service
UI

View history

Click the History tab to view history.

This screenshot shows the mobile work item form interface. At the top, a header reads "This is the new mobile experience!". Below the header, there are tabs: "Details", "History" (which is highlighted with an orange border), "Links (2)", and "Attachments". The main content area is divided into sections by expandable arrows: "Today", "Last seven days", and "Last thirty days". Each section lists activity items with icons, names, actions, and timestamps.

Section	Activity Item	Icon	Description	Timestamp
Today	Jamal Hartnett made field changes	User	2:20 PM	
	Jamal Hartnett added link	Link	2:10 PM	
	Jamal Hartnett added link	Link	2:10 PM	
▼ Last seven days				
Last seven days	Christie Church changed Value Area to Architectu...	User	Wed 5/17	
	Jamal Hartnett added comment @Chuck Reinhart - Note the differences betwe...	User	Wed 5/17	
▼ Last thirty days				
Last thirty days	Raisa-Pokrovskay changed Stack Rank to 10000...	User	5/4/2017	
	Raisa-Pokrovskay created the User Story	User	5/4/2017	

View and open work items in your activity lists

From within the mobile work item form, you can access your work items. The mobile browser allows you to view and open work items which fall into these categories:

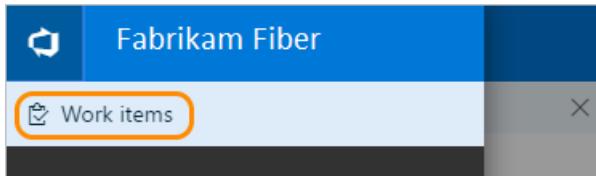
- **Assigned to me:** lists all work items assigned to you
- **Following:** lists all work items that you have elected to follow
- **My activity:** lists all work items that you have recently viewed or updated.

The lists available from each page span all team projects that you work in.

To access a list, first click the list control from the work item form you've opened.



Then, click **Work items**.



The browser opens to the **Assigned to me** page. From there, you can choose **Following** or **My activity** to access the other pages. To learn more about the **Work Items** view, see [View and add work items](#).

Work items

[Assigned to me](#) [Following](#) [My activity](#)

Doing

- 13 Secure sign-in
- 5 Make feedback mechanism mobile friendly
- 12 Change initial view
- 14 Interim save on long form
- 15 Welcome back page
- 16 Cancel order form
- 17 Resume
- 18 GPS locator interface
- 19 Review all security threats
- 11 Slow response on information form
- 1 This is the new mobile experience!

Related articles

Additional experiences are in the works to improve and expand on the mobile experience. For more information, see the blog post: [The mobile work item form \(preview\)](#).

- [Set personal notifications](#)
- [Set team notifications](#)
- [Follow a work item](#)

Provide feedback for the mobile experience

Help us improve the mobile experience.

To provide feedback, click the list control from the work item form and then click **Send Feedback**. To complete the feedback, select either the smile or frown and optionally enter a comment.



Work items

X



Jamal Hartnett
fabrikamfiber4@hotmail.com

Sign out

Send Feedback

View full site

Keyboard shortcuts for Azure DevOps and Team Explorer

9/4/2019 • 9 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

You can use the keyboard shortcuts listed in this article when you work within Azure DevOps or Team Explorer. In addition to these shortcuts, you can [assign your own shortcuts in Visual Studio](#) from the **Tools/Options/Environment/Keyboard** page.

For specific guidance on navigating within the web portal, see [Web portal navigation](#).

Web portal

You can use these keyboard shortcuts when working in the web portal for Azure DevOps.

Navigate	Navigate within lists
Ctrl+Alt,a Move focus to  admin link	Tab Move focus
Ctrl+Alt,h Move focus to  help link	  Move focus left/right
Ctrl+Alt,s Move focus to search box	  Move focus up/down
Ctrl+Alt,  Move focus to next section	Ctrl+Home Move focus to top of list
Ctrl+Alt,  Move focus to previous section	Ctrl+End Move focus to bottom of list
	Ctrl+↑ ↓ Move item up/down within list
	Shift↑ ↓ Highlight consecutive items
	Menu Open context menu
	Esc Dismiss context menu
	  Move focus left/right
	  Move focus up/down
	Enter Choose selected menu item

Web portal, global shortcuts

Type **?** to access the Global and page-specific shortcuts.

NOTE

The following shortcuts are available for TFS 2015.2 and later versions. Type **?** to access the Global and page-specific shortcuts.

You can use the following keyboard shortcuts from the web portal.

?

- ? Show shortcuts
- p Projects and teams

g,h Go to [Projects home](#)

g,w Go to [Boards or Work](#)

g,c Go to [Repos or Code](#)

g,b Go to [Pipelines or Build and release](#)

g,t Go to [Test Plans or Test](#)

g,s Go to [\[Project Settings\]](#)
(/azure/devops/organizations/settings/about settings)

f,n Focus next section

f,p Focus previous section

/ Move focus to search

Page-specific shortcuts only work when in a specific page. For example, type **g c** to open the **Code** page, and then type **c p** to create a pull request. These navigation shortcuts work as long as the focus is not on an input control.

Code

You can use the following keyboard shortcuts when working from a page under **Repos** or **Code**. To view the valid shortcuts, enter **Shift+?** to access Global and service-specific shortcuts..

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 and later versions.

Code

- r Select repository
- e Open explorer
- h Open history
- b Open branches (Git)
- q Open pull requests (Git)
- c,p Create pull request (Git)

c Open changesets (TFVC)

v Open shelvesets (TFVC)

File Explorer

- 1 Open contents
- 2 Open history
- t Move focus to directory path
- w Select branch (Git)
- y Switch to commit (Git)
- c,b Create branch (Git)

Work Items

You can use the following keyboard shortcuts when working from the **Repos>Work Items** or **Work>Work Items** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services.

Work

- l** Open [backlog](#)
- b** Open [board](#)
- i** Open [current iteration](#)
- t** Open [task board](#)
- q** Open [queries](#)
- z** Toggle full screen

Work items

- Ctrl+Shift+f** Filter results
- Ctrl+c** Copy to clipboard
- Delete** Delete

Work item form shortcuts

You can use the following keyboard shortcuts when [interacting with a work item form](#). To view the valid shortcuts, enter **Shift+?** from within the form.

NOTE

Feature availability: The following shortcuts are available from Azure DevOps Services.

Work Item Form

- Alt+i** Assign work item to me
- Ctrl+Shift+d** Go to discussion
- Ctrl+s** Save changes
- Shift+Alt+c** Copy work item title
- Ctrl+Shift+,** Move to left tab (page)
- Ctrl+Shift+.** Move to right tab (page)

Rich text field shortcuts

The rich text editor tool bar displays below the text entry area when you click your cursor within the each text box that can be formatted.

You can use the following keyboard shortcuts when working in a web browser on one of the following operating systems. (**Command**=

WINDOWS OS SHORTCUTS

Ctrl + b = Bold
Ctrl + c = Copy text
Ctrl + i = Italics
Ctrl + k = Insert hyperlink
Ctrl + s = Save
Ctrl + u = Underline
Ctrl + v = Paste text
Ctrl + y = Redo
Ctrl + z = Undo
Ctrl + . = Bullet list
Ctrl + / = Numbered list
Shift + : = Emoji library

MAC OS SHORTCUTS

Command + b = Bold
Command + c = Copy text
Command + i = Italics
Command + k = Insert hyperlink
Command + s = Save
Command + u = Underline
Command + v = Paste text
Command + Z = Redo
Command + z = Undo
Command + . = Bullet list
Command + / = Numbered list
Shift + : = Emoji library

The rich text formatting toolbar appears above each text box that can be formatted. It only becomes active when you click within the text box.



You can use the following keyboard shortcuts when working in the editor from a web browser running on a Windows operating system.

Ctrl + b = Bold
Ctrl + c = Copy text
Ctrl + i = Italics
Ctrl + k = Insert hyperlink
Ctrl + s = Save
Ctrl + u = Underline
Ctrl + v = Paste text
Ctrl + y = Redo
Ctrl + z = Undo
Ctrl + . = Bullet list
Ctrl + / = Numbered list
CTRL+Spacebar = Remove formatting

Backlogs

You can use the following keyboard shortcuts when working from a **Boards>Backlogs** page. These shortcuts work when you are on a product backlog, portfolio backlog, or sprint backlog page.

Backlogs

Ctrl+Home	Move item to top
m,b	Move item to backlog
m,i	Move item to current iteration
m,n	Move item to next iteration
Ins	Add child
Ctrl+Shift+f	Filter results

You can use the following keyboard shortcuts when working from a **Work>Backlogs** page. These shortcuts work when you are on a product backlog, portfolio backlog, or sprint backlog page.

	<p>Backlogs</p> <p>Ctrl+Home Move item to top</p> <p>m,b Move item to backlog m,i Move item to current iteration m,n Move item to next iteration</p> <p>n Open new item panel Ins Add child Ctrl+Shift+f Filter results</p> <p>r Show/Hide Parents</p>
--	---

Boards

You can use the following keyboard shortcuts from any Kanban board, that is, when working from **Repos>Boards** or **Work>Board** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 and later versions.

	<p>Kanban Board</p> <p>n Add new item c Add new child item</p> <p>Home Select first item Enter Open item</p> <p>Ctrl+Shift+f Filter results</p> <p>Ctrl+ [] Move item up Ctrl+ [] Move item down Ctrl+ [] Move item left Ctrl+ [] Move item right</p> <p>Ctrl+Home Move item to top of column Ctrl+End Move item to bottom of column Ctrl+Shift+ [] Move item to swimlane above Ctrl+Shift+ [] Move item to swimlane below</p> <p>F2 Rename item e Show/hide empty fields o Expand all swimlanes u Collapse all swimlanes</p> <p>Shift+Pageup Select first/next swimlane above Shift+Pagedown Select last/next swimlane below</p>
--	---

Queries

You can use the following keyboard shortcuts when [working with queries](#) in the web portal. To view the valid

shortcuts, enter **Shift+?** from **Boards>Queries** or **Work>Queries**.

NOTE

The following shortcuts are available from Azure DevOps Services or TFS 2015.2 or later versions.

Queries

c q Add new query

r or **Alt+r** Refresh query

Alt+q Return to query

j or **Alt+n** Select next item

k or **Alt+p** Select previous item

Ctrl+Shift+f Filter results

Plans

You can use the following keyboard shortcuts when [interacting with a delivery plan](#). To view the valid shortcuts, enter **Shift+?** when viewing a plan from the **Boards>Plans** or **Work>Plans** page.

NOTE

The following shortcuts are available from the web portal for TFS 2017.2 and later versions. Type **?** to access Global and service-specific shortcuts.

Work

Delivery plan

I Open backlog
b Open board
i Open current iteration
t Open task board
q Open queries
z Toggle full screen mode

Home Select first item
Enter Open item
n New item
Ctrl+ [] Move item up
Ctrl+ [] Move item down
Ctrl+ [] Move item left
Ctrl+ [] Move item right

Shift+ [] Pan timeline left

Shift+ [] Pan timeline right

u Collapse all backlogs

o Expand all backlogs

Shift+pageup Focus on previous team

Shift+pagedown Focus on next team

Ctrl+Shift+f Filter results

Test Plans, Parameters, and Runs

You can use the following keyboard shortcuts when working in **Test Plans** or **Test**.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2015.2 or later versions.

	<p>Test</p> <ul style="list-style-type: none"> n Open test plans m Open shared parameters r Open runs h Open machines
	<p>Test plan</p> <ul style="list-style-type: none"> 1 Open tests 2 Open charts e Execute tests t,b Mark selected tests as blocked t,f Fail selected tests t,n Mark selected tests as NA t,p Pass selected tests t,r Reset tests to active <p>Ctrl+Shift+f Filter results</p> <p>v,g View grid</p>
	<p>Parameters</p> <ul style="list-style-type: none"> 1 View parameter set grid 2 Open properties c,s Add parameter set c,t Add test case v,t Toggle test cases pane
	<p>Test runs</p> <ul style="list-style-type: none"> 1 Test runs 2 Filter

Wiki

NOTE

Keyboard shortcuts to manage Wiki pages are supported on TFS 2018.2 or later versions. To download TFS 2018.2, see [Team Foundation Server 2018 Update 2 Release Notes](#).

You can use the following keyboard shortcuts when [managing or editing Wiki pages](#). To view the valid shortcuts, enter **Shift+?** from a **Wiki** page.

NOTE

The following shortcuts are available from the web portal for Azure DevOps Services and TFS 2018.2 and later versions.

	Wiki (manage)
	n Add new page e Edit page c Add new sub-page
	Ctrl+↑ Move page up the order Ctrl+↓ Move page down the order
	Ctrl+P Print page Ctrl+Shift+f Filter page
	Wiki edit
	Ctrl+b Bold text Ctrl+i Italicize text Ctrl+k Insert hyperlink Ctrl+c Copy text Ctrl+v Paste copied text
	Ctrl+Shift+f Format tables
	Ctrl+s Save changes Ctrl+Enter Save and Close Esc Close

Team Explorer navigational shortcuts

Use these shortcuts when working in Team Explorer.

Navigate	Context menu
Ctrl+0,a Open web portal	Open a context menu
Ctrl+0,b Open Build	Esc Dismiss a context menu
Ctrl+0,c Open Connect	Move focus left/right
Ctrl+0,d Open Documents	Move focus up/down
Ctrl+0,e Open Branches (Git)	Enter Choose Context menu
Ctrl+0,g Open Changes (Git)	
Ctrl+0,h Open Home	
Ctrl+0,m Open My Work (TFVC)	
Ctrl+0,p Open Pending changes (TFVC)	Work item commands
Ctrl+0,r Open Reports	
Ctrl+0,s Open Settings	
Ctrl+0,w Open Work items	
Ctrl+0,y Open Synchronization (Git)	
Ctrl+' Move focus to search box	Alt+m,g Open work item
Alt+0 Move focus to top of page	Alt+m,i Add a work item
Alt+1...9 Move focus to visible section [1 thru 9]	Alt+m,q Add a query
Alt+↑↓ Move focus to next/previous section	Shift+Alt,c Copy selected work item
	Shift+Alt,l Link to new work item
	Enter Open selected work item

You can use query results shortcuts whenever you have a list of work items, such as the query results view or a list of linked work items within a work item form.

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
	Move focus left/right		Scroll left/right
	Move focus up/down	PgUp/PgDn	Scroll up/down

QUERY EDITOR	ACTION	QUERY RESULTS	ACTION
Shift+ []	Highlight consecutive clauses	Shift+ []	Highlight consecutive rows
Shift+ []	Move focus left one field at a time	Shift+Alt,n	Move focus to next item
Shift+ []	Move focus right one field at a time	Shift+Alt,p	Move focus to previous item
End	Move focus to end of current clause	End	Move focus to bottom of list
Enter	Move focus down	Enter	Open selected work item
Tab	Move focus right, one field at a time	Home	Move focus to top of list
Ctrl+c	Copy selected clause	+/-	Expand/collapse current row
Ctrl+s	Save changes (editor)	Ctrl+s	Save changes (results)
Ctrl+v	Paste copied clause	F5	Refresh
Del	Delete contents of current field or clause		

Related articles

- [Keyboard shortcuts for Microsoft Test Manager](#)
- [Customize Visual Studio keyboard shortcuts](#)
- [Default keyboard shortcuts for Visual Studio](#)
- [Accessibility Features of Visual Studio](#)
- [Web portal navigation](#)

Install Team Explorer

Team Explorer is a plug-in to Visual Studio. By installing the free [Visual Studio Community](#), other Visual Studio version, or Visual Studio Team Explorer 2017 you gain access to Team Explorer.

Learn more about [working in Team Explorer](#).

Search

8/1/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Search makes it easy to locate information across all your projects, from anywhere and any computer or mobile device, using just a web browser.

5-Minute Quickstarts

Search code	Search work items
Search packages	Search wiki

Videos

<https://channel9.msdn.com/Events/Visual-Studio/Connect-event-2015/500/player>

How-to Guides

- [Use Code Search](#)
- [Use Work Item Search](#)
- [Use Wiki Search](#)
- [Use Package Search](#)
- [Administer Search](#)

Resources

- [Adhoc vs managed work item queries](#)
- [Search your Wiki](#)
- [Code Search blog posts](#)
- [Code Search on Marketplace](#)
- [Work Item Search blog posts](#)

Search across all your code, wiki, packages and work items

8/1/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Azure DevOps and TFS store a tremendous amount of information; work items for planning and tracking your work, wiki for sharing information, version control to track code file changes, and build and release services to help you ship effortlessly. As your backlog and codebase expand and are divided across multiple projects, teams and repositories, finding what you need becomes increasingly difficult. To maximize cross-team collaboration and sharing, you need solutions that can quickly and efficiently locate information across all your projects. The Search extensions available for Azure DevOps and TFS enable you to search across all the projects, teams and repositories to which you have access.

The screenshot displays two search interfaces side-by-side. The left interface is for 'Code' and the right is for 'Work items'. Both interfaces share a common header with a search bar (highlighted with a red box), filter icons, and user profile options.

Code Search Interface:

- Filter by field:** Options include 'Assigned To', 'Created By', 'State', and 'Work Item Type'.
- Filter by scope:** Options include 'With file extension', 'Filename', 'Under path', 'Inside project', and 'Inside repository'.
- Filter by code type:** Options include 'Argument', 'Basetype', 'Caller', 'Class', 'ClassDeclaration', 'ClassDefinition', 'Comment', 'Constructor', 'Declaration', 'Definition', and 'Show more'.
- Operators:** Examples of operators shown are AND, NOT, and OR.
- Help:** A link to the 'help page'.

Work item Search Interface:

- Filter by field:** Options include 'a: Assigned To', 'c: Created By', 's: State', and 't: Work Item Type'.
- Filter by scope:** Options include 'ext: With file extension', 'file: Filename', 'path: Under path', 'proj: Inside project', and 'repo: Inside repository'.
- Filter by code type:** Options include 'arg: Argument', 'basetype: Basetype', 'caller: Caller', 'class: Class', 'classdecl: Class Declaration', 'classdef: Class Definition', 'comment: Comment', 'ctor: Constructor', 'decl: Declaration', 'def: Definition', and 'Show more'.
- Operators:** Examples of operators shown are AND, NOT, and OR.
- Help:** A link to the 'help page'.

Both interfaces show a list of search results on the right, each with a preview and an ellipsis (...).

The Search extensions make it easy to search for information across all your projects, from anywhere and any

computer or mobile device, using just a web browser. You can narrow down your results and focus in on what you need by using filters.

- [Code Search](#)
- [Work Item Search](#)
- [Wiki Search](#)
- [Package Search](#)

See also:

- [Adhoc vs managed work item queries](#)
- [Search your Wiki](#)

Search your code

9/4/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Use Code Search to search across all of your projects, find specific types of code, and easily drill down or widen your search.

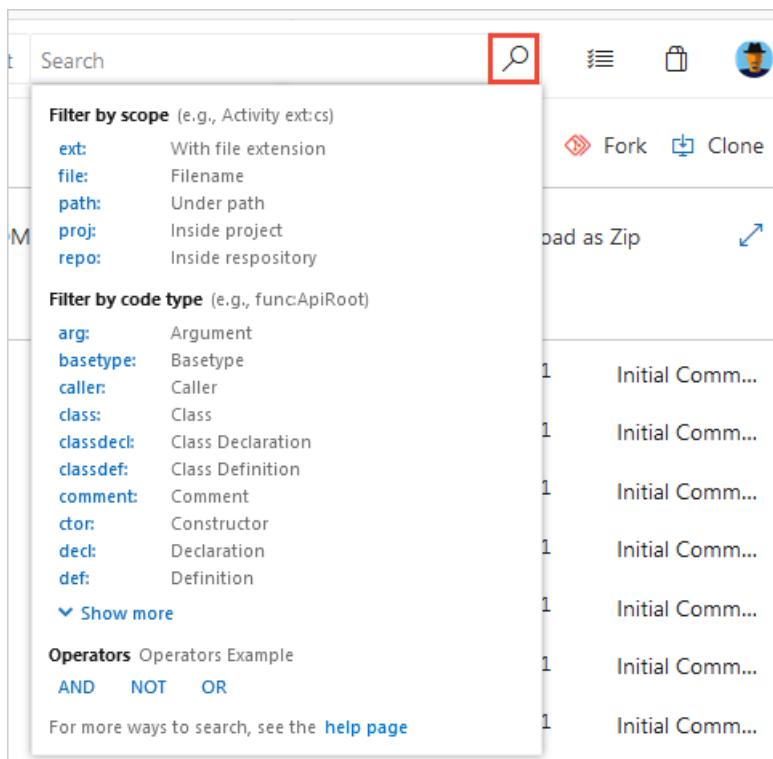
Prepare

Go to [Visual Studio Marketplace](#) to install the extension as an administrator. Non-administrative users can also request the extension is added. For more information, see [Install an extension](#).

Only users with Basic access can use Code Search.

Start searching

1. Open the **Azure Repos** section in Azure DevOps (see [Web portal navigation](#)).
2. Choose the  icon at the top right of the window to show the search textbox.

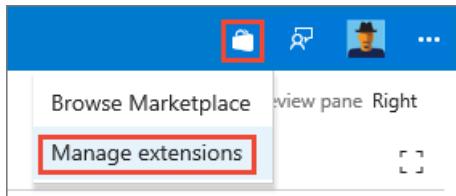


3. Enter a search string in the textbox, and press *Enter* (or choose the  icon) to start your search.

1. In the search textbox at the top right of the window, check that the text says *Search code*. The search text box may say *Search work items*. In this case, use the drop-down selector to change it.



If you don't see the  selector in the Search box, open the **Manage extensions** page and check that Code Search is installed ([see Administer Search](#)).



2. Enter a search string in the textbox, and press *Enter* (or choose the icon) to start your search.

View the results

1. The search page shows a list of the matching code files. The selected file has all instances of the search string highlighted (only the first 100 hits are highlighted).

This project ViewResult

Code 2 Work item 0 Wiki 0

Repo: PartsUnlimited Branch: master Path: /

Showing 2 code results

File	Project	Branch	Matches
OrdersControllerTests.cs	PartsUnlimited > PartsUnlimited	master	12 matches
HomeControllerTests.cs	PartsUnlimited > PartsUnlimited	master	2 matches

OrdersControllerTests.cs

Contents History Compare Blame

```
35     var viewResult = resultTask.Result;
36
37     // act
38     var resultTask = await controller.Orders.GetAsync();
39     var viewResult = resultTask.Result;
40
41     // assert
42     Assert.IsNotNull(viewResult);
43     mockOrdersQuery.Verify(o => o.Id == 1);
44     var model = viewResult.Model;
45     Assert.IsNotNull(model);
```

If you see a list of work items, ensure that **Code** is selected in the top left.

2. Sort the results as you need using the drop-down list of properties, or by relevance.

Path: / Code Type: All

Sort by

Relevance
File Path
File Name

```
1 using Microsoft.VisualStudio.TestTools.UnitTesting;
2 using Moq;
```

Open the search results in a new browser tab from either search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the icon. In Google Chrome and Firefox press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

3. Try assembling more complex search strings using the operators and functions listed in the handy drop-down list. Select the filter function or code type you want to include in your search string from the list. Then, enter the criteria value.

Filter by scope (e.g., Activity ext:cs)

- ext:** With file extension
- file:** Filename
- path:** Under path
- proj:** Inside project
- repo:** Inside repository

Filter by code type (e.g., funcApiRoot)

- arg:** Argument
- basetype:** Basetype
- caller:** Caller
- class:** Class
- classdecl:** Class Declaration
- classdef:** Class Definition
- comment:** Comment
- ctor:** Constructor
- decl:** Declaration
- def:** Definition

Show more

Operators Operators Example
AND NOT OR

For more ways to search, see the [help page](#)

- You can find all instances of "ToDo" comments in your code simply by selecting `comment:` and typing `todo`.
- You can search in specific locations, such as within a particular path, by using a search string such as `Driver path:MyShuttle/Server`.
- You can search for files by name, such as `Driver file:GreenCabs.cs`, or just by file extension. For example, the search string `error ext:resx` could be useful when you want to review all error strings in your code. But, even if your plain text search string (without specific file-type functions) matches part of a filename, the file appears in the list of found files.
- You can combine two or more words by using Boolean operators; for example, `validate OR release`.
- You can find an exact match to a set of words by enclosing your search terms in double-quotes. For example, `"Client not found"`.
- You can use the code type search functions with files written in C#, C, C++, Java, and Visual Basic.NET.
- See also [full details of the search syntax](#).

4. Widen your search to all projects or your entire organization. Or narrow it to specific areas and types of code by selecting from the drop-down lists at the top of the page.

The screenshot shows the search interface for an organization named 'PartsUnlimited'. The search bar at the top is labeled 'Search this organization' and has a red box around it. Below the search bar, there are three main filter sections:

- Repo:** A dropdown menu set to 'PartsUnlimited' with a red box around it. Other options include 'All Repos' and 'PartsUnlimited' (which is checked).
- Branch:** A dropdown menu set to 'master' with a red box around it. Other options include 'Configure searchable branches' and 'PartsUnlimited-aspNet45'.
- Code Type:** A dropdown menu set to 'All' with a red box around it. Other options include 'All Code Types', 'Declaration' (1 result), 'Definition' (1 result), and 'Reference' (2 results).

The results pane shows a list of files under 'PartsUnlimited' with a red box around the first item, 'ControllerTests.cs'. The list includes file names, paths, match counts, line numbers, and snippets of the code.

5. Use the tabs in the results page to view the history of the file and to compare versions of the file.

OrdersControllerTests.cs

Contents History Compare Blame

↑ ↓ ⏪ ⏫

9f81fbc1 ▾ ↔ 58d9b870 ▾

26 -		var fakeDb = new FakeDataContext	26 +		var mockDb = new MockDataContext	26 +	
27 -		var fakeModel = new OrdersModel	27 +		var mockModel = new OrdersModel	27 +	
28			28			28	
29 -		var fakeOrdersQuery = new Mock<	29 +		var mockOrdersQuery = new Mock<	29 +	
30 -		fakeOrdersQuery.Setup(o => o.In	30 +		mockOrdersQuery.Setup(o => o.In	30 +	
31 -		.ReturnsAsync(fakeModel);	31 +		.ReturnsAsync(mockModel);	31 +	
...							

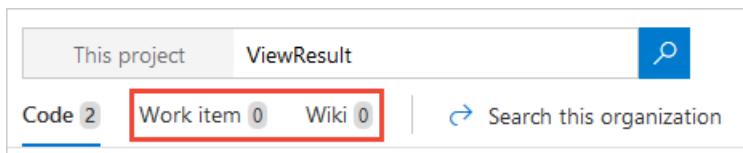
6. Choose the filename link at the top of this column to open the file in a new Code Explorer window.

OrdersControllerTests.cs

Contents History Compare Blame

↑ ↓ ⏪ ⏫

7. Quickly [search for work items](#) containing the same search string, or search for the same string in your project's wiki.



How to search filters on a Multi Repo Multi branch environment

1. Enter search text `NOT kjhasdhkjashdkjkhjdashkjdsaahsdkj` and hit enter.
2. Choose project filter as Team Project Name. Choose repository filter as your GIT repo. Choose branch filter as the wanted branch.
3. Enter search text `ext:json` and hit enter. You can see your wanted results.

This search is a generic text, which will have matches in all repositories like NOT `kjhasdhkjashdkjkhjdashkjdsaahsdkj`. The long string could be any garbage text, which won't be present in any file. Adding a NOT before it inverts the logic and hence matches all files.

Next step

[Learn more about Code Search](#)

Search for work items

9/4/2019 • 2 minutes to read • [Edit Online](#)

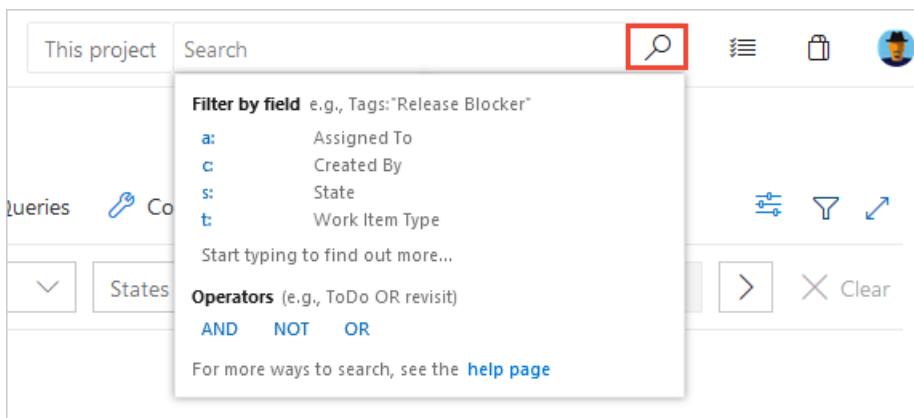
Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

Work Item Search provides fast and flexible search across all your work items over all your projects.

See also: [Adhoc vs managed work item queries](#)

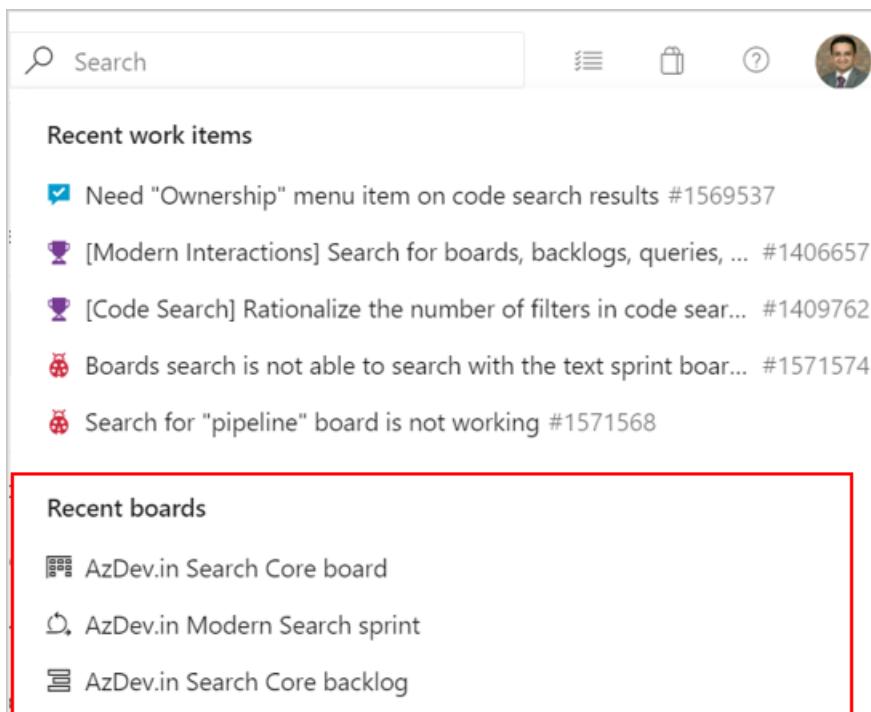
Start searching

1. Open the **Azure Boards** section in Azure DevOps (see [Web portal navigation](#)).
2. Choose the  icon at the top right of the window to show the search textbox.



3. Enter a search string in the textbox, and select *Enter* (or choose the  icon) to start your search.

Access your recently visited boards, backlogs, queries, and sprints.



The screenshot shows the search results page with a red box around the "Recent boards" section. The "Recent work items" section lists several items with checkmarks and icons, such as "Need 'Ownership' menu item on code search results #1569537" and "[Modern Interactions] Search for boards, backlogs, queries, ... #1406657". The "Recent boards" section lists three boards: "AzDev.in Search Core board", "AzDev.in Modern Search sprint", and "AzDev.in Search Core backlog".

You can also search for the boards, backlogs, queries and sprints across your project by entering the board name in the search box.

The screenshot shows the Microsoft Azure DevOps search interface. At the top, there is a search bar with the text "Atlas". To the right of the search bar are several icons: a magnifying glass, a dropdown arrow, a folder, a question mark, and a user profile picture. Below the search bar, the results are categorized into sections:

- Work items**
 - PowerBI connector GA Signoff from Atlas #1428619
 - Clean up/reset Atlas-Interview #1553146
 - Action Required : Feature flags inconsistency between Scale... #1559364
 - [Atlas] Move production related release definitions from \Te... #1511521
 - Action Required : Feature flags inconsistency between Scale... #1547721
- Other items in Boards**
 - Atlas backlog
 - Atlas board
 - Atlas sprint
- Others**
 - View code files (50+)
 - View wiki pages (10)

1. In the search textbox at the top right of the window, check that the text says *Search work items*.



2. If you have the Code Search extension installed, the search text box may say *Search code*. In this case, use the drop down selector to change it.
3. Enter a search string in the textbox, and press *Enter* (or choose the icon) to start your search.

1. In the search textbox at the top right of the window, check that the text says *Search work items*.



2. If you have the Code Search extension installed, the search text box may say *Search code*. In this case, use the drop-down selector to change it.
3. Enter a search string in the textbox, and press *Enter* (or choose the icon) to start your search.

View the results

1. Search results are displayed in a snippet view, where matches found are shown in bold.

This project login

Code 0 Work item 3 Wiki 0 Search this organization Relevance View

Area: FabrikamFiber Web Types: All

Showing 3 work item results Provide feedback

Work Item ID	Title	Status	Reason	Area	Iteration
62	Login page	Resolved		FabrikamFiber W	
119	Login behaviour for booking	Closed	Description: Login behaviour for booking	FabrikamFiber W	
97	Login and logout behaviours	Closed	Description: Login and logout behaviours	FabrikamFiber W	

This is a full text search that uses simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Searches are *not* case-sensitive.

2. Select a snippet of a work item to display it in the right window. You can edit and manage this work item in the usual way.

USER STORY 62

62 Login page

Save Follow ...

Blocked X Xamarin X +

State: Resolved Area: FabrikamFiber Web
Reason: Code complet ... Iteration: FabrikamFiber Web\Iteration 1

Description

Login page

- New linked work item
- Change type...
- Move to team project
- Create copy of work item...
- Email work item
- Delete
- Templates >
- Start storyboarding
- Request feedback
- New branch...
- Customize
- Keyboard shortcuts
- Add to dashboard >

Open the search results in a new browser tab from a search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the icon. In Google Chrome or Firefox, press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

3. Fine-tune your search by specifying the fields to search. Enter and a user name to search for all items assigned to that user.

Search work items 🔍

Filter by field e.g., Tags:"Release Blocker"

a: Assigned to
c: Created by
s: State
t: Work item type

Start typing to find out more...

Operators e.g., a:@me (t:Task OR t:Bug)

AND NOT OR

For more ways to search, see the [help page](#)

The quick filters you can use are:

- **a:** for **Assigned to**:
- **c:** for **Created by**:
- **s:** for **State**
- **t:** for **Work item type**

4. Start typing the name of a field in your work items; for example, type **ta**.

ta 🔍

Filter by field e.g., Tags:"Release Blocker"

Tags
Target
Target Date
Target Resolve Date
Task Due Date
Task Type

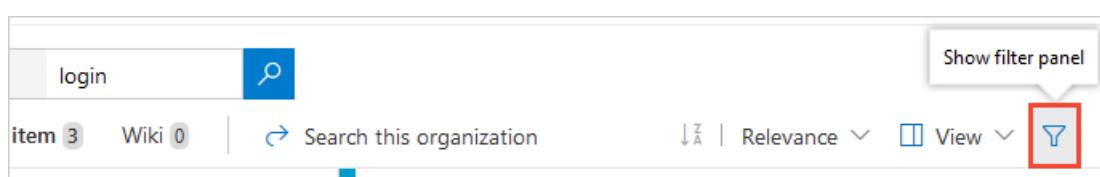
Operators e.g., a:@me (t:Task OR t:Bug)

AND NOT OR

For more ways to search, see the [help page](#)

The dropdown list shows work item field name suggestions that match user input, helping the user search faster. For example, a search such as **tags:Critical** finds all work items tagged 'Critical'.

5. Add more filters to further narrow your search, and use Boolean operators to combine terms, if necessary.
For example, **a: Chris t: Bug s: Active** finds all active bugs assigned to a user named Chris.
6. Widen your search across all projects, or narrow it to specific types and states. Use the "filter" icon to show the selector lists.



login 🔍

Show filter panel

item 3 Wiki 0 🔍 Search this organization Relevance View Filter icon (highlighted with a red box)

7. Select the criteria you want in the drop-down selector lists, or search across the entire organization.

The screenshot shows the 'Work item' search results for the 'FabrikamFiber Web' project. The search bar at the top contains 'login'. Below the search bar, there are three tabs: 'Code 0', 'Work item 3' (which is selected), and 'Wiki 0'. A red box highlights the 'Search this organization' button. On the left, a tree view shows 'FabrikamFiber Web' expanded, with 'Team1' and 'Team2' as children. A red box highlights the 'X →' button next to 'FabrikamFiber Web'. To the right, there are two dropdown menus: 'Types: All' (selected 'All Types') and 'States: All' (selected 'All States'). Both dropdowns have red boxes around them. Under 'Types: All', there are checkboxes for 'User Story' (unchecked) and 'All Types' (checked). Under 'States: All', there are checkboxes for 'Closed' (unchecked) and 'Resolved' (unchecked).

8. Sort the results as you need using the drop-down list of field names, work item types, or by relevance.

The screenshot shows the 'Work item' search results for the 'FabrikamFiber Web' project. The search bar at the top contains 'login'. Below the search bar, there are three tabs: 'Code 0', 'Work item 3' (selected), and 'Wiki 0'. A red box highlights the 'Relevance' dropdown in the top right corner. On the left, a tree view shows 'FabrikamFiber Web' expanded, with 'Team1' and 'Team2' as children. A red box highlights the 'X →' button next to 'FabrikamFiber Web'. The search results list three items: '62 Login page', '119 Login behaviour for booking', and '97 Login and logout behaviours'. To the right, a 'Sort by' dropdown menu is open, with 'Relevance' selected. Other options include 'Assigned To', 'Changed Date', 'Created Date', 'ID', 'State', 'Tags', 'Title', and 'Work Item Type'. A red box highlights the 'Relevance' option in the dropdown menu.

9. Quickly [search for code](#) containing the same search string, or search for the same string in your [project's wiki](#).

The screenshot shows the 'Work item' search results for the 'FabrikamFiber Web' project. The search bar at the top contains 'login'. Below the search bar, there are three tabs: 'Code 0' (selected), 'Work item 3', and 'Wiki 0'. Each tab has a red box around it. A red box also highlights the 'Search this organization' button.

Next step

[Learn more about Work Item Search](#)

Search your packages

9/27/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

Package Search provides fast and flexible search across all your packages inside your organization.

Learn more about packages here: [Azure Artifacts in Azure DevOps Services](#)

Start searching

1. Open the **Azure Artifacts** section in Azure DevOps (see [Web portal navigation](#)).
2. Choose the  icon at the top right of the window to show the search textbox.



3. Enter a search string in the textbox, and press *Enter* (or choose the  icon) to start your search.

View the results

1. Search results are displayed with matches to user query shown in bold.

This project windowsazure 

Code 50 Work item 50 Wiki 27 Package 120

Feeds: All Views: Type: All

Showing 25 of 120 package results

- MSNuGet-backup > WindowsAzure**
WindowsAzure .NET SDK 2.2
- Codex-Deps > WindowsAzure.Storage**
This client library enables working with the Microsoft Azure storage services which include the blob and file service for storing binary and text data, the table service for storing structured non-relational data, and the queue service for storing messages that may be
- VS > WindowsAzure.Storage**
This client library enables working with the Microsoft Azure storage services which include the blob and file service for storing binary and text data, the table service for storing structured non-relational data, and the queue service for storing messages that may be
- Domino-Selfhost > WindowsAzure.Storage**
This client library enables working with the Microsoft Azure storage services which include the blob and file service for storing binary and text data, the table service for storing structured non-relational data, and the queue service for storing messages that may be

This is a full text search that uses simple search strings for words or phrases. Note that searches are *not* case-sensitive.

Open the search results in a new browser tab from a search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the  icon. In Google Chrome or Firefox, press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

2. Widen your search across all feeds, or narrow it to specific views and package types. Note that Views filter would only appear if a single feed is selected from Feeds filter. Use the "filter" icon to show the selector lists.

This screenshot shows the package search interface. At the top, there is a search bar with the text "This project package". Below the search bar are navigation links: "Code 50", "Work item 50", "Wiki 50", "Package 40K+", and a "Show filter panel" button. A red box highlights the "Show filter panel" button.

3. Select the criteria you want in the drop-down selector lists, or search across the entire organization.

This screenshot shows the package search interface with filters applied. The search bar now contains "package". The filter panel is open, showing dropdowns for "Feeds: ApplicationInsigh..." (set to "All"), "Type: All" (set to "All"), and a "Clear" button. A red box highlights the "Type: All" dropdown. The main search results show a list of feeds, with "ApplicationInsigh..." selected. Other feeds listed include "AsimLongFeedNa...", "APEX", "ApexPreRelease", "APEXSRE", and "AppAnalyticsRC". The results are paginated with "Showing 177 feeds".

4. By switching pivots, quickly search [code](#) containing the same search string, or search for the same string in your [wikis](#) or search for the same string in your [work items](#)

This screenshot shows the package search interface with different pivots selected. The search bar now contains "packagesearch". The pivots at the bottom are: "Code 22" (highlighted with a red box), "Work item 50" (highlighted with a red box), "Wiki 4" (highlighted with a red box), and "Package 0". A red box highlights the "Code" pivot.

Next step

[Learn more about Package Search](#)

How To: Use Code Search

9/27/2019 • 10 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

By using Code Search you can:

- **Search across all of your projects:** Search in your own codebase and your partner teams' codebases. Use cross-project searches over all the code in your Azure DevOps or TFS instance to search across your enterprise's entire codebase. Narrow your search by using project, repository, path, file name, and other filter operators. Use wildcards to widen your search and Boolean operators to fine-tune it.
- **Find specific types of code:** Use code type filters to search for specific kinds of code such as definitions, references, functions, comments, strings, namespaces, and more. You can use Code Search to narrow down your results to exact code type matches. Navigate quickly to a method definition to understand its implementation simply by applying the definition filter, or scope the search to references in order to view calls and maximize code reuse.
- **Easily drill down or widen your search:** When you find an item of interest, simply place the cursor on it and use the shortcut menu to quickly search for that text across all your projects and files. Easily trace how your code works by using the shortcut menu to search for related items such as definitions and references - directly from inside a file.

NOTE

You cannot search code in forked repositories.

Syntax for simple and compound searches

Use simple search strings for words or phrases. The default is a whole word search; for example, a search for "valid" will not find instances of the word "validation". However, searches are *not* case-sensitive.

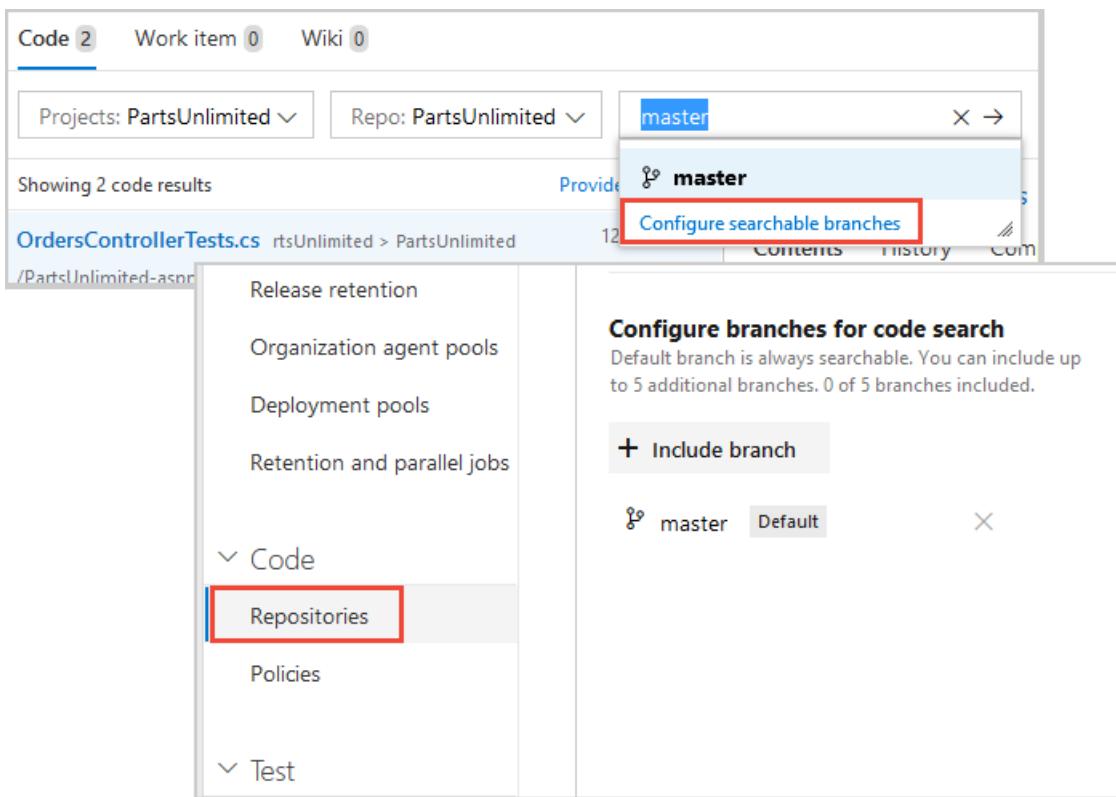
Words separated by spaces, and not wrapped in double-quotes, are treated as separate search terms and the search will expect to find an occurrence of all the words (in other words, it assumes the `AND` operator between words).

You must escape the special characters `(`, `)`, `[`, `]`, `:`, `*`, and `?` by enclosing them in a phrase delimited with double-quotes `"` and `"`.

When you search from inside a project, the default is to search only within that project. You can navigate up to higher levels to broaden your search.

In a Git project, you see a list of the repositories it contains. Use the project and repository checkboxes to widen your search to more or all projects, or to narrow your search to fewer projects and repositories. If there is more than a few projects or repositories, use the **Show more** link to see them all.

Code Search can index multiple branches in a Git repository - by default it indexes files in only the default branch of your Git repositories (usually this is the **master** branch). Specify the branches for each repository that you want Code Search to index in the **Options** tab of the **Repositories** section in the [project settings page](#).



In a TFVC project, you see a list of folder paths in that project for which you have read access - you won't see any projects and folders for which you don't have read permission. Select paths in the folder tree to narrow your search if required.

Code Search remembers your last settings, such as the project and repository or path you searched in. Clear all the checkboxes to search across all projects. Do this quickly and easily with the **Clear all** links when you want to search in a different scope.

In the results pane, Code Search highlights up to the first 100 hits or matches found in the target files.

Narrow your search by using Boolean operators

Narrow your search by using Boolean operators to combine search criteria. Combine multiple search criteria using **AND**, **OR**, or **NOT** (they must be uppercase).

Use parentheses to specify the precedence of the operations when you use more than one Boolean operator. By default, a search combines all the words you enter using the **AND** operator so that it will return only files that contain all of the words you entered.

For example:

- `validate AND revisit` finds files that contain both the words **validate** and **revisit**. Note that **AND** is the default operator, and so this is equivalent to the search string `validate revisit`.
- `validate OR revisit` finds files that contain either of the words **validate** or **revisit**.
- `validate NOT revisit` finds files that contain the word **validate** but not the word **revisit**.
- `(validate NOT revisit) OR "release delayed"` finds files that contain the word **validate** but not the word **revisit** or files that contain the phrase **release delayed**.

Broaden your search by using wildcards

Use the wildcard characters `*` and `?` to broaden your search criteria. For example:

- `CodeSenseHttp*` finds files containing words that start with **CodeSenseHttp**, such as **CodeSenseHttpClient** and **CodeSenseHttpClientTest**.
- `CodeA23?R` finds files containing words that start with **CodeA23**, have any alphanumeric character next, and end with **R**. For example, **CodeA234R** and **CodeA23QR**.

You can use wildcard characters anywhere in your search string **except** as a **prefix** in a simple search string or a query that uses a [code type filter](#). For example, you cannot use a search query such as `*RequestHandler` or `class:?RequestHandler`. However, you can use prefix wildcards with the other search filter functions; for example, the search query strings `file:*RequestHandler.cs` and `repo:?Handlers` are valid.

You can use more than one `?` wildcard to match more than one character.



No results found for ...

If there is a very large number of hits when using a wildcard search, such as when using a very simple wildcard search string, you may see a message that no matching files were found. In this case, narrow your search to reduce the number of matches. For example, specify more characters of the word(s) you want to find, or add a condition or filter to limit the number of possible matches.

Search for phrases

To find an exact match to a set of words, enclose your search terms in double-quotes to perform a *phrase search*. For example, `"Client not found"`.

Within a phrase:

- Boolean operators are treated as literal text.
- The query language characters `:()[]*?` are treated as literal text.
- You need to escape only the special characters `\` and `"`.

Search for special characters

NOTE

This feature is currently available only for Azure DevOps Services customers and not TFS.

You can include special characters in a search string, or search specifically for special characters, according to the following rules:

- Search for any special character that is not a part of the query language, (for example, excluding the characters `:()[]*?`) as either a simple search string or a phrase search string. For example, `react-redux` or `"react-redux"` will produce the same results.
- Search for a special character that is a part of the query language (`:()[]*?`) by enclosing the search string within double-quotes. For example, `"flatten()` will find the literal string `flatten()`.
- Search for a literal occurrence of the double-quote character `"` by preceding it with the escape character `\` and enclosing the search string in double-quotes. For example, `"\"react-redux\""` will find the literal string `"react-redux"`.

Functions to find specific types of code

As you type in the search box, select functions and keywords from the drop-down list to quickly create your query. Use the **Show more** link to display all the available functions and keywords. Mix and match the functions as required.

You can also select one or a combination of filters from the list in the left column. Again, the **Show more** link displays all the available functions and keywords.

Alternatively, you can type the functions and parameters directly into the search box. The following table shows the full list of functions for selecting specific types or members in your C#, C, C++, Java, and Visual Basic.NET

code.

TO FIND CODE WHERE <i>FINDTHIS</i> APPEARS AS A SEARCH FOR ARGUMENT ARG: <i>FINDTHIS</i>
Argument	arg: <i>findThis</i> <small>Deprecated in July 2019</small>
Base type	basetype: <i>findThis</i>
Calling function	caller: <i>findThis</i> <small>Deprecated in July 2019</small>
Class definition or declaration	class: <i>findThis</i>
Class declaration	classdecl: <i>findThis</i> <small>Merged with class:</small>
Class definition	classdef: <i>findThis</i> <small>Merged with class:</small>
Comment	comment: <i>findThis</i>
Constructor	ctor: <i>findThis</i> <small>Merged with method:</small>
Declaration	decl: <i>findThis</i>
Definition	def: <i>findThis</i>
Destructor	dtor: <i>findThis</i> <small>Merged with method:</small>
Enumerator	enum: <i>findThis</i>
Extern	extern: <i>findThis</i> <small>Deprecated in July 2019</small>
Field	field: <i>findThis</i>
Friend function	friend: <i>findThis</i> <small>Deprecated in July 2019</small>
Function	func: <i>findThis</i> <small>Merged with method:</small>
Function declaration	funcdecl: <i>findThis</i> <small>Merged with method:</small>
Function definition	funcdef: <i>findThis</i> <small>Merged with method:</small>
Global	global: <i>findThis</i> <small>Deprecated in July 2019</small>
Header	header: <i>findThis</i> <small>Deprecated in July 2019</small>
Interface	interface: <i>findThis</i>
Macro	macro: <i>findThis</i>
Macro definition	macrodef: <i>findThis</i> <small>Merged with macro:</small>
Macro reference	macroref: <i>findThis</i> <small>Merged with macro:</small>

TO FIND CODE WHERE <code>FINDTHIS</code> APPEARS AS A SEARCH FOR ARGUMENT ARG: <code>FINDTHIS</code>
Method	<code>method:findThis</code>
Method declaration	<code>methoddecl:findThis</code> Merged with method:
Method definition	<code>methoddef:findThis</code> Merged with method:
Namespace	<code>namespace:findThis</code>
Property	<code>prop:findThis</code>
Reference	<code>ref:findThis</code>
String literal	<code>strlit:findThis</code>
Struct	<code>struct:findThis</code> Merged with type:
Struct declaration	<code>structdecl:findThis</code> Merged with type:
Struct definition	<code>structdef:findThis</code> Merged with type:
Template argument	<code>tmplarg:findThis</code> Deprecated in July 2019
Template specification	<code>tmplspec:findThis</code> Deprecated in July 2019
Type	<code>type:findThis</code>
Typedef	<code>typedef:findThis</code> Merged with type:
Union	<code>union:findThis</code> Deprecated in July 2019

Functions to select projects, repositories, paths, and files

Functions make it easy to narrow the search to specified locations, specific types of files within these locations, or specified filenames. Mix and match the functions as required.

Narrow the search to a specific location using the `proj`, `repo`, or `path` filters:

- `QueueJobsNow proj:Fabrikam` finds all occurrences of the word **QueueJobsNow** in the **Fabrikam** project.
- `QueueJobsNow repo:Contoso` finds all occurrences of the word **QueueJobsNow** in the **Contoso** repository.
- `QueueJobsNow path:VisualStudio/Services/Framework` finds all occurrences of the word **QueueJobsNow** in the path **VisualStudio/Services/Framework** and its sub-paths.
- Enclose the argument to the filter in double-quotes if it contains a space. For example:
`QueueJobsNow path:"VisualStudio/Windows Phones and Devices/Services"`.

Narrow the search to specific files using the `file` or `ext` filters:

- `QueueJobsNow file:queueRegister*` finds all occurrences of the word **QueueJobsNow** in all files where the filename starts with **queueRegister**.
- `QueueJobsNow ext:cs` finds all occurrences of the word **QueueJobsNow** in only C# source files.
- A plain text search string that does not include file type functions will also find files where the string matches

part of the filename.

Find related items or other terms

One of the powerful features of Code Search is the capability to expand your search interactively, based on the results of previous searches. For example, you can easily broaden your search to related files when tracing or debugging code.

Place the insertion point on a term in the file and open the shortcut menu (mouse: right-click) to start a new search for other files containing the selected term. You can search for it as text, for the definition if you select an object name, or for references to a selected object.

More examples

Some more examples of search strings are:

- You can find all instances of "ToDo" comments in your code simply by selecting `comment: todo`.
- You can search in specific locations, such as within a particular path, by using a search string such as `Driver path:MyShuttle/Server`.
- You can search for files by name, such as `Driver file:GreenCabs.cs`, or just by file extension. For example, the search string `error ext:resx` could be useful when you want to review all error strings in your code. But even if your plain text search string (without specific file type functions) matches part of a filename, the file appears in the list of found files.
- You can combine two or more words by using Boolean operators; for example, `validate OR release`.
- You can find an exact match to a set of words by enclosing your search terms in double-quotes. For example, `"Client not found"`.
- You can use the code type search functions with files written in C#, C, C++, Java, and Visual Basic.NET.

Open the search results in a new browser tab from either search box by pressing *Ctrl + Enter* or by holding *Ctrl* and clicking the  icon. In Google Chrome, press *Ctrl + Shift + Enter* to switch the focus to the new browser tab.

Got feedback?



Report any problems on [Developer Community](#), or send feedback to vstssearch@microsoft.com.

How To: Use Work Item Search

8/1/2019 • 7 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017

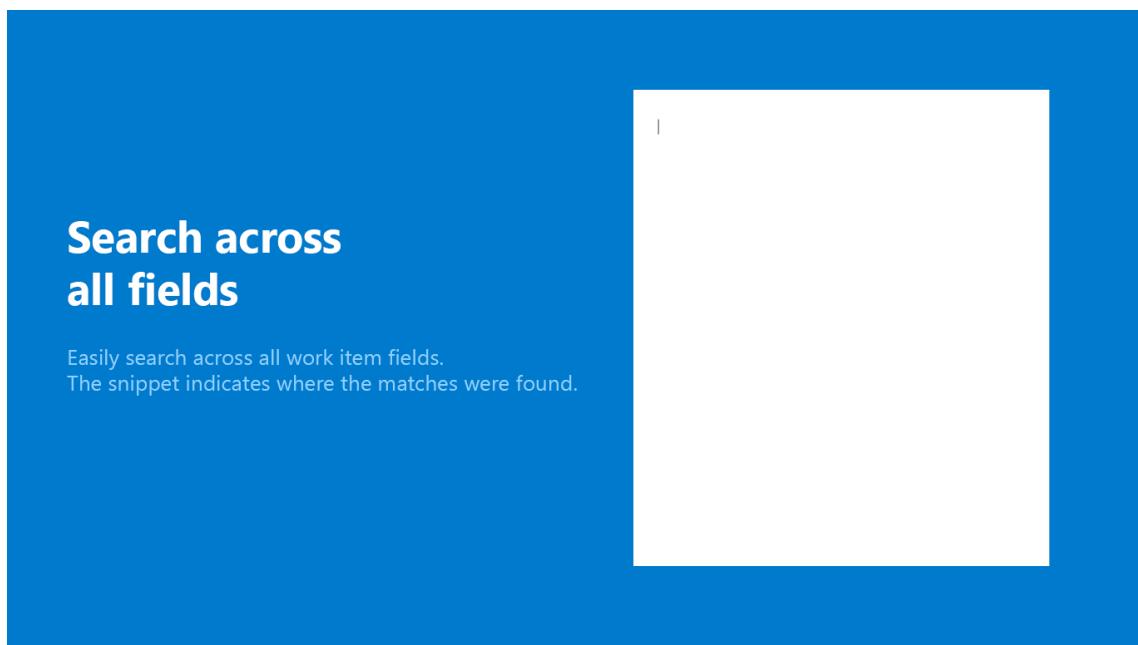
Work Item Search is available as a built-in extension in Azure DevOps and TFS. Users can use Work Item Search by default without any installation when Boards service is installed and enabled in Azure DevOps Services.

By using Work Item Search you can:

- **Search over all your projects:** Search in your own and your partner teams' backlog. Use cross-project searches over all the work items to search across your enterprise's entire work items. Narrow your search by using project and area path filters.
- **Search across all work item fields:** Quickly and easily find relevant work items by searching across all work item fields (including custom fields). Use a full text search across all fields to efficiently locate relevant work items. The snippet view indicates where matches were found.
- **Search in specific fields:** Use the quick in-line search filters, on any work item field, to narrow down to a list of work items in seconds. The dropdown list of suggestions helps complete your search faster. For example, a search such as **AssignedTo:Chris WorkItemType:Bug State:Active** finds all active bugs assigned to a user named Chris.
- **Take advantage of integration with work item tracking:** The Work Item Search interface integrates with familiar controls for managing your work items; letting you view, edit, comment, share, and much more.

Full text search across all fields

You can easily search across all work item fields, including custom fields, which enables more natural searches. The snippet view indicates where matches were found.



Use simple search strings for words or phrases. Work item search matches derived forms of your search terms; for example, a search for "updating" will also find instances of the word "updated" and "update". Note that searches are *not* case-sensitive.

When you search from inside a project, the default is to search only within that project. When you search from inside a team, the default is to search only within the default area path of that team. When you have one project selected, you see a list of area paths in that project for which you have read access - you won't see any projects and area paths for which you don't have read permission. Select area paths in the tree to narrow your search if required.

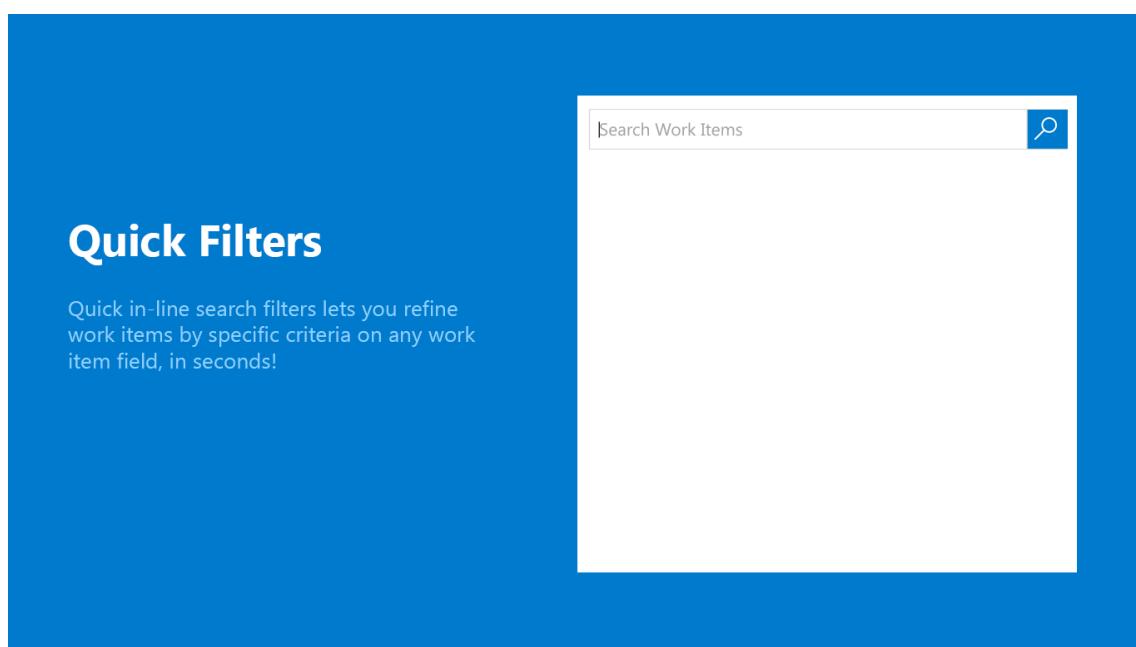
The selected projects are always at the top of the list. Notice that hit counts are also shown for projects that are not selected.

Open the search results in a new browser tab from either search box by pressing *Ctrl+Shift+Enter* or by holding *Ctrl* and clicking the  icon.

Quick Filters for matching in specific fields

Quick inline search filters let you refine work items in seconds. The dropdown list of suggestions helps complete your search faster. Mix and match the functions to create quick powerful searches. For example:

- Scope your search terms to match in any work item field including custom fields. Simply type the field name followed by the search terms; for example, a search such as **tags:Critical** finds work items having a field 'tags' containing the term 'Critical'.
- Use multiple inline search filters to scope your search by any work item field, including custom fields. For example, a search such as **t: Bug path:"project\search"** finds all bugs in the area path "project\search".
- Use the operators `>`, `>=`, `<`, `<=`, `=`, and `!=` for date, integer and float fields. For example, a search such as **t: Bug CreatedDate> @Today-7** finds all bugs created in the last week
- For the search query that contain multiple terms and users looking for exact match, embed the search term inside " "; for example, a search such as **BuildPath: "tools.demoproject.com"** finds all work items that necessarily contain the path "tools.demoproject.com".



Quick Filters shortcuts

The common inline search filters can be quickly accessed using shortcuts:

- `a:` for **Assigned to**:
- `c:` for **Created by**:
- `s:` for **State**

- `t:` for **Work item type**

For example, you can use quick searches such as `a:@Me s:active t:bug` to find all bugs assigned to you.

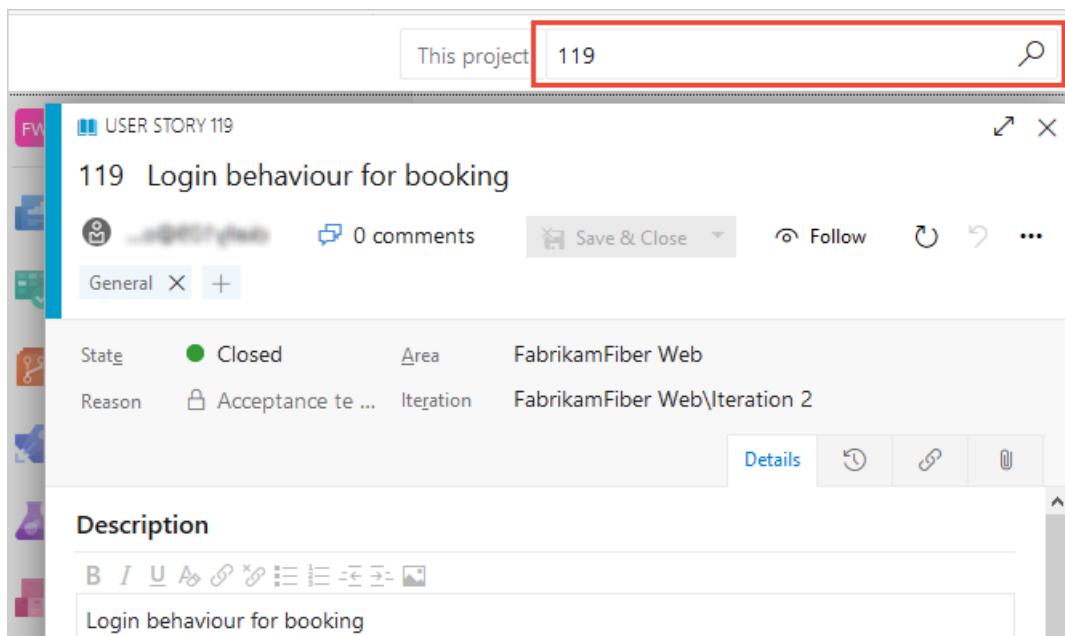
Filters to scope projects, area and iteration paths

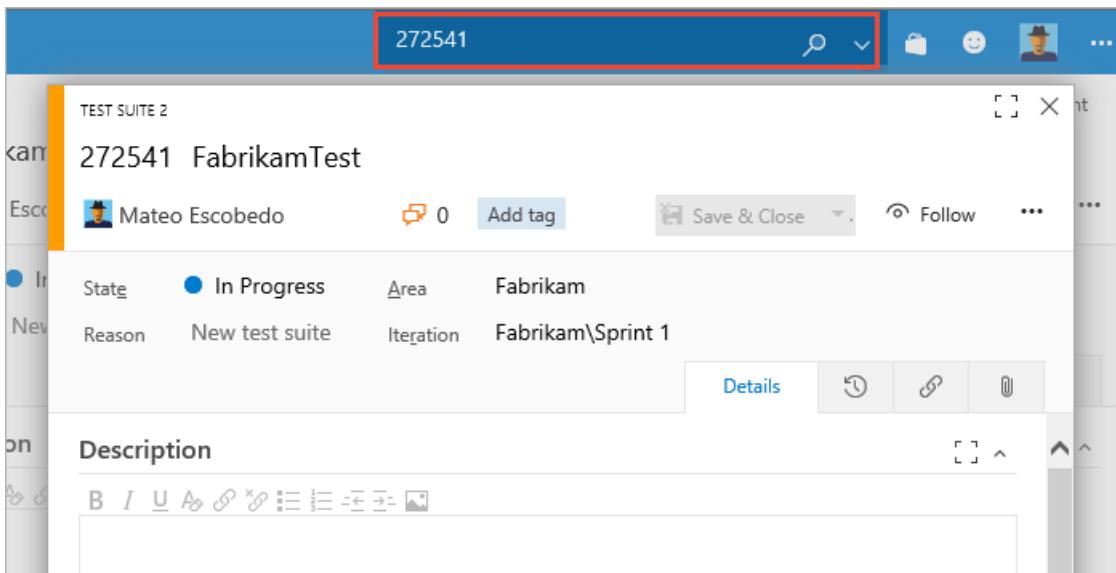
Filters make it easy to narrow the search to specified projects and area paths. Narrow the search to a specific location using the `proj`, `area`, or `iteration` filters:

- `Wiki proj:Fabrikam` finds all occurrences of the word **Wiki** in the **Fabrikam** project.
- `Wiki area:Contoso/Mobile` finds all occurrences of the word **Wiki** in the area path **Contoso/Mobile** and its sub-paths.
- `Wiki iteration:Contoso/Sprint101` finds all occurrences of the word **Wiki** in the iteration path **Contoso/Sprint101** and its sub-paths.
- Enclose the argument to the filter in double-quotes if it contains a space. For example:
`Wiki path:"Contoso/Windows Phones and Devices/Services"`.

Quickly navigate to a work item using its ID

Type or paste the work item ID in the search box in the Azure DevOps or TFS title bar to quickly navigate to it. Searching for a work item ID opens the work item in a modal dialog, providing quick access to read and edit work items.





Narrow your search with Boolean operators

Narrow your search by using Boolean operators to combine search criteria. Combine multiple search criteria using `AND`, `OR`, or `NOT` (they must be uppercase).

Use parentheses to specify the precedence of the operations when you use more than one Boolean operator. By default, a search combines all the words you enter using the `AND` operator so that it will return only work items that contain all of the words you entered.

For example:

- `welcome AND page` finds work items that contain derived forms of both the words **welcome** and **page**. Note that `AND` is the default operator, and so this is equivalent to the search string `welcome page`.
- `signup OR signin` finds work items that contain either of the words **signup** or **signin**.
- `signin NOT signup` finds work items that contain the word **signin** but not the word **signup**.
- `(signin NOT signup) OR "user login"` finds work items that contain the word **signin** but not the word **signup** or work items that contain the phrase **user login**.

To find an exact match to a set of words, enclose your search terms in double-quotes. For example,

`"Client not found"`

Broaden your search with wildcards

Use the wildcard characters `*` and `?` to broaden your search criteria. For example:

- `Browser*` finds work items containing words that start with **Browser**, such as **BrowserEdge**, **BrowserIE** and **BrowserFirefox**.
- `alpha?version` finds work items containing words that start with **alpha**, have any alphanumeric character next, and end with **version**. For example, **alpha1version** and **alphaXversion**.

You can use wildcard characters anywhere in your search string **except** as a **prefix**. For example, you cannot use a search query such as `*RequestHandler`. However, you can use prefix wildcards with the other search filter functions; for example, the search query strings `area:*mobile` and `tags:*Browser` are valid.

You can use more than one `?` wildcard to match more than one character.



No results found for ...

- If there are no results matching the input, try removing filters and retry the search. This will broaden the search and after you view the search results, you can apply appropriate filters again and search again for relevant results
 - Check for the spelling of your search terms. Currently Work item search doesn't support ignoring of users' spelling mistakes
 - If there are a very large number of hits when using a wildcard search, such as when using a very simple wildcard search string, you may see a message that no matching files were found. In this case, narrow your search to reduce the number of matches. For example, specify more characters of the word(s) you want to find, or add a condition or filter to limit the number of possible matches.
-

See more of the work item

You can quickly get a full screen view of the selected work item using the expand  and shrink  icons in the toolbar. However, another way to see more of the work item, while still being able to select work items from the list of matching results, is to hide the left column filter pane by choosing the < icon at the top left of the column. Use the > icon to restore the filter pane.

If you are using a portrait orientation screen, use the **Preview pane: Right** link at the top right of the window to display the code below the search results list.

Search remembers the state of the filter pane, configuration of the work item view pane, and its position between sessions as part of your user preferences.

Got feedback?



Report any problems on [Developer Community](#), or send feedback to vstssearch@microsoft.com.

How To: Use Package Search

9/27/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

Package Search is automatically available to users of Azure DevOps Services. By using Package Search you can:

- **Search package by title and description:** You can quickly and easily find relevant packages by using free text search on title and description. You can also narrow your search by using Boolean operators and combine search criteria.

[Web portal package search](#)

- **Search across all of your organization feeds:** Search in your and your team's feeds across the organization. Narrow your search by using feed, view and package type filters. Use wildcards to widen your search and Boolean operators to fine-tune it.

[Web portal package search across organization feeds](#)

Syntax for simple and compound searches

Use simple search strings for words or phrases. The default is a whole word search; for example, a search for "config" will not find instances of the word "configuration". However, searches are *not* case-sensitive.

Words separated by spaces, and not wrapped in double-quotes, are treated as separate search terms and the search will expect to find an occurrence of all the words (in other words, it assumes the `AND` operator between words).

Irrespective of which project you search from, the default is to search within all feeds of the organization.

Narrow your search by using Boolean operators

Narrow your search by using Boolean operators to combine search criteria. Combine multiple search criteria using `AND`, `OR` or `NOT` (they must be uppercase).

Use parentheses to specify the precedence of the operations when you use more than one Boolean operator. By default, a search combines all the words you enter using the `AND` operator so that it will return only files that contain all of the words you entered.

For example:

- `xml AND parser` finds packages that contain both the words **xml** and **parser**. Note that `AND` is the default operator, and so this is equivalent to the search string `xml parser`.
- `xml OR parser` finds packages that contain either of the words **xml** or **parser**.
- `xml NOT parser` finds packages that contain the word **xml** but not the word **parser**.
- `(xml NOT parser) OR lib` finds packages that contain the word **xml** but not the word **parser** or packages that contain the word **lib**.

Broaden your search by using wildcards

Use the wildcard character `*` and `?` to broaden your search criteria. For example:

- `xmlparser*` finds packages containing words that start with **xmlparser**, such as **xmlparsersdk** and

xmlparserlib.

- `mypackage?` finds packages containing words that start with **mypackage** and have any one alphanumeric character next, such as **mypackage1**, **mypackage2** and **mypackage3**

You can use more than one wildcard to match more than one character.

Search for phrases

To find an exact match to a set of words, enclose your search terms in double-quotes to perform a *phrase search*.

For example, `"package for markup files"`.

Within a phrase:

- Boolean operators are treated as literal text.

Got feedback?



Report any problems on [Developer Community](#), or send feedback to vstssearch@microsoft.com.

Set up and administer Code Search, Wiki Search, and Work Item Search

10/1/2019 • 27 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#)

In this topic:

- **Configure Search (Code, Work Item, and Wiki) in Azure DevOps Services**
 - [Install the Search extension in Azure DevOps Services](#)
 - [Uninstall the Search extension from Azure DevOps Services](#)
- **Configure Search (Code, Work Item, and Wiki) in Azure DevOps Server or Team Foundation Server (TFS)**
 - [Install the Search extension in Azure DevOps Server or TFS](#)
 - [Configure Search in Azure DevOps Server or TFS](#)
 - [Secure Search in Azure DevOps Server and TFS](#)
 - [Upgrade Search in Azure DevOps Server and TFS](#)
 - [Manage Search in Azure DevOps Server and TFS](#)
 - [Uninstall Search from Azure DevOps Server or TFS](#)
 - [Limitations of Search in Azure DevOps Server and TFS](#)
 - [Troubleshoot Search in Azure DevOps Server and TFS](#)

Also see [Install and configure Azure DevOps Server or TFS](#) and [Requirements and compatibility](#).

Users with at least a **Basic** access can use Code Search. **Stakeholders** do not have access to code, and therefore no access to Code Search. All users have access to Work Item and Wiki Search.

Install the Search extension in Azure DevOps Services

- For Work Item and Wiki search, there is no installation required because these are built-in features of Azure DevOps Services.
- For Code Search, go to [Azure DevOps Marketplace](#) to install the Code Search extension in Azure DevOps as an administrator. Non-administrative users can also go here to request the Code Search extension be added to Azure DevOps Services.

For more details, see [Install an extension](#) in the Marketplace documentation.

Uninstall the Search extension from Azure DevOps Services

- For Work Item and Wiki search, users cannot uninstall them as these come as built-in extensions in Azure DevOps Server and TFS.
- For Code Search, see [Uninstall or disable an extension](#) in the Marketplace documentation.

Install Search extension in Azure DevOps Server or TFS

Availability

- Code Search is available in TFS 2017 and later, and in Azure DevOps Server.
- Work Item Search is available in TFS 2017 Update 2 and later, and in Azure DevOps Server.
- Wiki Search is available in TFS 2018 Update 2 and later, and in Azure DevOps Server.
- Work Item and Wiki search are built-in extensions that are installed by default during Search configuration.

Installation

Code Search is an opt-in feature, and can be installed later from the Local Gallery. To do this, go to **Local Gallery** (http://{Server}/tfs/_gallery) as an administrator. Non-administrative users can also go here to request the extension be added to TFS or Azure DevOps Server.

For more details, see [Install an extension](#) in the Local gallery documentation.

Configure Search in Azure DevOps Server or TFS

Configure the Search service using the dedicated pages in the Server Configuration Wizard as you install Azure DevOps Server or TFS. You can also [configure and unconfigure Search](#) afterwards by running the Server Configuration Wizard again or by launching the Search Configuration Wizard.

Hardware recommendations

Search can be used on any size physical server or virtual machine that runs TFS 2017 or later, or Azure DevOps Server. It can be configured on the same server, or on a separate server dedicated to Search. When configuring Search on the same server, you must consider the existing CPU utilization factor due to Azure DevOps Server or TFS itself.

For production environments we recommend you configure Search on a separate server.

For acceptable performance in multi-user scenarios, consider the following recommendations:

- Less than 250 users with Search co-located on the server (typically used for demonstration and trial purposes):
 - Quad core processor, 8 GB (minimum) RAM
 - CPU Utilization factor less than 50%
 - Fast hard drive backed by Solid State Drive (SSD) storage
- Less than 500 users with Search located on a [separate server](#):
 - Dual core processor, 8 GB (minimum) RAM
 - Fast hard drive backed by Solid State Drive (SSD) storage
- Less than 1,000 users with Search located on a [separate server](#):
 - Quad core processor, 16 GB (minimum) RAM
 - Fast hard drive backed by Solid State Drive (SSD) storage
- More than 1,000 users with Search located on a [separate server](#):
 - Quad core processor, 16 GB (minimum) RAM
 - Fast hard drive backed by Solid State Drive (SSD) or Storage Area Network (SAN) storage
- Azure DevOps Server with Multiple ATs:
 - Install Search on a [separate server](#)
- Azure DevOps Server CPU utilization greater than 50% before installing Search:

- Install Search on a [separate server](#)

Disk space requirement:

The amount of disk space taken up by Search depends mainly on the type and size of files that will be indexed. For Code search, since many times repositories can be large and can have different code files in version control, disk space requirement could be significant. As a general guideline, allocate up to 150% of the size of all the repositories that will be indexed. From TFS 2018 Update 3 onwards and Azure DevOps Server, users can exclude desired folders from their repositories during the time of indexing if they want to optimize the disk space consumed by search.

Software Dependencies

Search has the following dependencies, which are installed automatically as part of the configuration:

- [Elasticsearch](#) by Elasticsearch BV (see Notes 1 and 2)
- [Elasticsearch NEST client](#)
- [Azul Zulu OpenJDK](#) (see [Java installation notes](#))
- [Markdowndeep](#) by Topten Software
- [Roslyn](#) compiler platform
- [ANTLR](#) language recognition parser

NOTES:

1. Search uses a modified version of Elasticsearch. It will work only with this modified version.
2. A newer version of Elasticsearch ships with TFS 2018 Update 2 and above, and Azure DevOps Server. Upgrading from an older version of Search will result in all content being re-indexed after the installation. Depending on the volume of content (code files, work items, and wiki pages), re-indexing can take some time to complete.
3. The system or server administrator must ensure that Server JRE is maintained and updated in line with the software provider's recommendations. Also see the [Java installation notes](#) that follow.
4. The Azul Zulu OpenJDK does not automatically install updates. Ensure you regularly [check for updates](#).

Java installation notes

If the Search configuration wizard does not detect a working installation of a Java Runtime Environment (JRE), it provides an option to download and install the latest supported version. Internet connectivity is required to download this. If the target server does not have Internet connectivity, you must download and install a JRE manually before attempting to install Search.

Versions of Search prior to Azure DevOps Server used the [Oracle Server Java Runtime Environment](#). In Azure DevOps Server, the default JRE is [Azul Zulu OpenJDK](#).

During installation, the wizard sets the **JAVA_HOME** environment variable to point to the JRE installation folder. The configuration wizard may fail to detect an existing JRE installation if it is not correctly configured, or if the **JAVA_HOME** setting points to an earlier version than that required by Search.

NOTE

We don't advise installing Elasticsearch on a machine where resources are shared, especially on a large enterprise environment with multiple application tiers. Instead, we recommend setting up Elasticsearch in a separate dedicated machine. In that way, the JAVA environment isn't shared across machines for other purposes.

If there is a version of a JRE **earlier** than the minimum required by Search, and the **JAVA_HOME** variable is set to that version, we recommend you install Search on a separate server because changing the value of the **JAVA_HOME** variable may cause other installed software to fail.

If there is a version of Server JRE **equal to or later** than the minimum required by Search, and it is not recognized by the configuration wizard, you must set the value of the **JAVA_HOME** variable to that version as described in the JRE installation guide and then rerun the configuration wizard.

- [Zulu OpenJDK installation guide](#)
- [Oracle JRE troubleshooting guide](#)

If you cannot install the version of Java required by Search due to other dependencies, you can:

- Install Azure DevOps Server or TFS together with the Search extension on a server that does not have Java installed (not recommended for more than 250 users or CPU utilization greater than 50% or multiple ATs).
- Install Search and the JRE on a [separate server](#) from Azure DevOps Server or TFS.

NOTE

If you are using Oracle Server JRE 8, which was the default for Search in TFS (Azure DevOps Server does not use Oracle Server JRE 8), be aware that:

- Search does not use or support any of the commercial features of Server JRE 8. Therefore, during Search configuration, the commercial features of the Server JRE are neither activated nor unlocked.
- If you choose to continue with Oracle JRE, contact Oracle for a [Java SE Subscription](#), so that you can continue to receive JRE updates.

Migrating to Zulu OpenJDK from Oracle Server JRE

Search in Azure DevOps Server and TFS supports both Azul Zulu OpenJDK and Oracle JRE, allowing you to choose between them based on your needs. When selecting a JRE during installation, Azure DevOps Server defaults to Azul Zulu OpenJDK 8.

To change to the Azul Zulu OpenJDK, follow these simple steps:



More details are available [here](#).

NOTE

- If you choose to use Azul Zulu OpenJDK, ensure you [download the latest updates](#). It does not automatically install updates.

Installation considerations

Consider the following when configuring Search:

- Both Work Item and Wiki search are enabled by default when Search is configured. These extensions can be later removed if required from the **Manage Extensions** page of Azure DevOps Server or TFS.
- The Code Search extension must be installed for each Azure DevOps Server or TFS collection where you want to use it. When initially configuring Search, you can set a checkbox to **Automatically install Code Search extension for existing and new Project Collections** to automate this process.
- If you do not set the checkbox to install the Code Search extension for all your project collections when configuring Search, your project collection administrator can install it from the Local Gallery. Ensure you navigate to the Local Gallery (http://{Server}/tfs/_gallery) from your Azure DevOps Server or TFS

portal page.

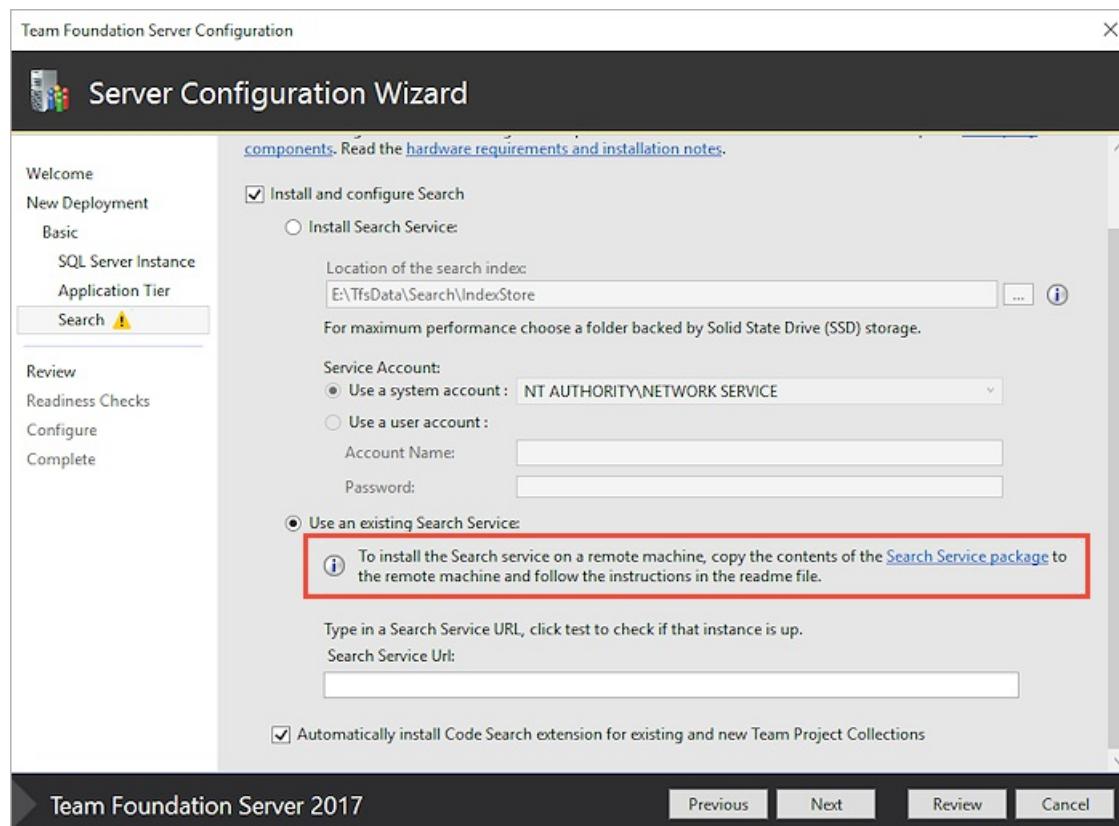
- The search index folder should be located on a separate fast hard drive backed by fast storage such as a solid-state drive (SSD) or Storage Area Network (SAN) to maximize search performance. As a general guideline, allocate up to 150% of the size of all the repositories that will be indexed. That is the worst-case scenario; the actual space consumed is dictated by the amount and type of code files, amount of work items and wiki pages in that collection.
- Unless specified, the indexing service and Elasticsearch engine use the network service account during installation to create and access the index files. If you choose a different account, it must have **Log on as a service** permission.
- Restrict the permissions for the index disk and folder to protect the index from accidental or malicious modification or deletion, and configure appropriate **security settings** for the service.
- When configuring Search for a server with **multiple application tiers (ATs)**, make sure it is installed on a **separate remote server**. After Search is installed on the remote server, use the Configuration Wizard on any one of the AT servers to link the remote Search instance with your Azure DevOps Server or TFS instance. When unconfiguring Search in the future you must use the Configuration Wizard on the same AT server where configuration was originally carried out.
- If you are performing a **pre-production upgrade** on a server where Search is already configured, you must fully reconfigure Search again to avoid corrupting your production instance. For this reason, there is no option to configure Search as part of a pre-production upgrade. Instead, configure it after the pre-production upgrade is complete. As this is a pre-production upgrade, you can choose to uncheck **Automatically install and configure Code Search for all existing and new collections** during configuration, and instead install the Search extension for just one or two of your collections after configuration is complete.
- If you are performing a **production upgrade** on a server where Search is already configured, and want to retain the Search feature, you must set the checkbox to **Install and Configure Search**. At this point the wizard will detect your existing Search instance and automatically select the **Use existing Search instance** option and pre-populate your current Search service URL. Use the **Install a new Search instance** option only if you want to set up a new instance of Search on the same server. Setting up a new instance causes all your code, work item and wiki to be indexed again, which - depending on the size of the collections - can take some time. During indexing, users may see partial search results.
- If you are **upgrading your server to new hardware**, depending on how Search was previously configured, you have two options:
 - If Search was configured on a separate server from Azure DevOps Server or TFS, you must select **Install and Configure Search** in the Server Configuration Wizard, and subsequently select **Use an existing Search instance** and provide the URL of your existing Search instance to complete the Search configuration.
 - If Search was configured alongside your Azure DevOps Server or TFS instance on the old server, you must select **Install and Configure Search** in the Server Configuration Wizard, and subsequently select **Install a new Search instance** again on the new server if you want to continue to co-host Search and Azure DevOps Server or TFS. This will cause all Search indexes for all collections to be re-created which, depending on the size of each collection, might take some time.
- If you are **detaching a collection** from one Azure DevOps Server or TFS instance in order to attach it to another instance, follow the below sequence:
 - Detach the collection from source Azure DevOps Server or TFS instance.
 - Configure Search on the target Azure DevOps Server or TFS instance (if not yet done already).

- Attach the collection to the target Azure DevOps Server or TFS instance
- Uninstall all the Search extensions (Code, Work Item, or Wiki) for that collection from the **Local Gallery** page of your source Azure DevOps Server or TFS instance.
- Install the Search extension (Code, Work Item, or Wiki) for that collection from the **Local Gallery** by browsing to it from your target Azure DevOps Server or TFS instance.

Installing or updating Search on a separate server

To install or update Search on a separate (remote) server, typically when you have more than 250 users, follow these steps:

1. As you install Azure DevOps Server or TFS on the primary server, set the **Install and configure Search** checkbox in the **Search** page of the Server Configuration Wizard.
2. Select the option to **Use an existing Search service**.
3. Use the **Search service package** link provided in the wizard to access a set of Search installer files on the local machine, and then copy these files to the remote server.

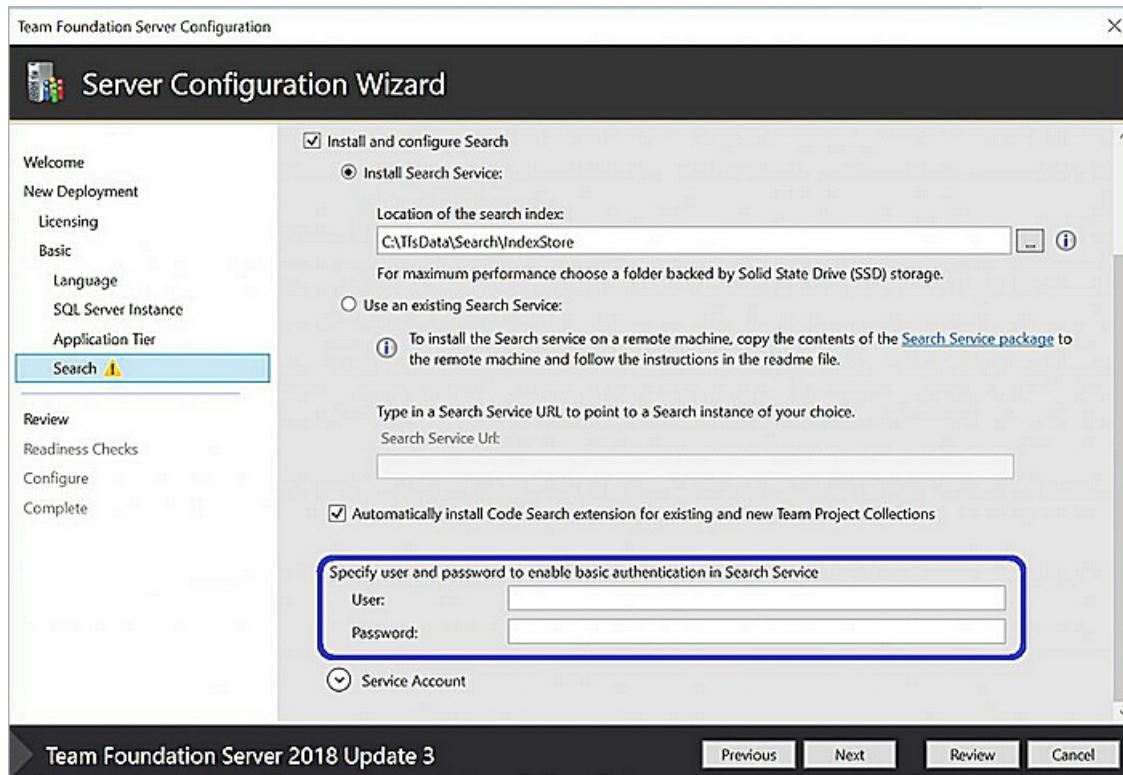


4. Follow the instructions in the **Readme.txt** file located in the set of installer files to install or update the Search service on the remote server.
5. After the installation of the Search service on the remote server is complete, copy the resulting Search server URL into the **Search URL** field of the configuration wizard running on the Azure DevOps Server or TFS instance.
6. When both installations are complete, configure appropriate **security settings** for both servers.

Secure Search in Azure DevOps Server and TFS

The Search service uses a modified version of **Elasticsearch** (the terms "Search" and "Elasticsearch" are used interchangeably for the remainder of this section). TFS Admins will need to provide credentials as part of configuring Search feature (through Server or Search configuration wizard) whether Search service is on the same machine as Azure DevOps Server or TFS, or on a separate machine. These are new set of credentials (not related

to any pre-existing account or server credentials) and are used to set up and connect to Search service. These new set of credentials will enable basic authentication in search service.



In the case of an upgrade from TFS 2018 Update 1.1 to TFS 2018 Update 3, or search re-configuration, only the user information is auto-populated and administrators must provide password credentials. Administrators have an option to provide different username and password if they wish. If the Search service is on the same machine as Azure DevOps Server or TFS, and administrators want to change the credentials, they can simply provide a new set of credentials in the Configuration Wizard to set up the Search service. However, if the Search service is on a remote machine, administrators must first provide the new credentials to the Search service setup script.

NOTE

- Username and password values should both be between 8 and 64 characters in length. While the password can be assigned any value, the username can contain only alphanumeric and underscore characters.
- Search credentials will only authenticate the users and make sure that unauthenticated users cannot access the Elasticsearch endpoint. However, Elasticsearch does not support HTTPS and so these credentials are sent over the network as Base64 encoded strings. If there is a possibility of intermediate access to request, configure appropriate security settings based on your corporate security and compliance requirements.
- Irrespective of the Azure DevOps Server or TFS version you use, aim to limit access to both searching and indexing to specific users or user groups using encryption through IPSec, as described below.

Consider the following techniques for using IPSec to secure Elasticsearch on a Windows server:

- **Configure security with authentication only:**

- This ensures only authorized users can access the Elasticsearch port. It requires only service-side rules (firewall rules on only the server running Elasticsearch).
- Prerequisite: Azure DevOps Server or TFS must be configured with a domain account.
- Follow the steps in [Creating Firewall Rules that Allow IPsec-protected Network Traffic](#).

- **Configure security with authentication, integrity protection, and encryption:**

- This ensures encryption and integrity protection are applied along with authentication. It requires both

client-side and service-side rules (firewall rules on the server running Elasticsearch and all Azure DevOps Server or TFS App Tier servers).

- Prerequisite: Azure DevOps Server or TFS must be configured with a domain account.
- Follow the steps in [Isolating a Server by Requiring Encryption and Group Membership](#).

Upgrade Search in Azure DevOps Server and TFS

TFS 2017 Update 1 includes updated Search components. If the Search service was configured in TFS 2017 RTM then, during an upgrade, the Search service components will be updated automatically if the Search service was configured on the TFS that is being upgraded. If Search was configured on a remote server, follow [these instructions](#) to update it.

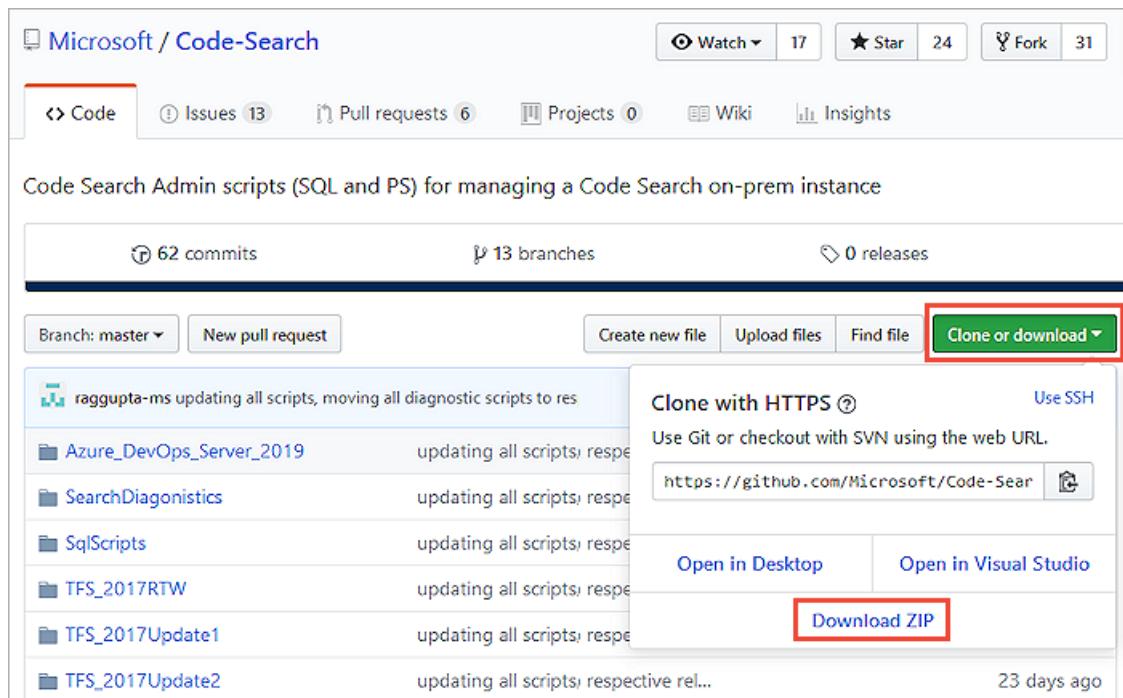
TFS 2017 Update 2 includes Work items Search. It uses the same Search service as Code Search. If the Search service was configured in TFS 2017 RTM/Update1 then, during an upgrade, the Search service components will be updated automatically if the Search service was configured on the TFS that is being upgraded. If Search was configured on a remote server, follow [these instructions](#) to update it.

TFS 2018 Update 2 includes updated Search components and Wiki Search. If the Search service was configured in TFS 2017 RTM, Update1, Update2, or TFS 2018 RTM then, during an upgrade, the Search service components will be updated automatically if the Search service was configured on the TFS that is being upgraded. If Search was configured on a remote server, follow [these instructions](#) to update it. In both cases, all existing content (code files and work items) will be automatically re-indexed to support the updated components after configuration. Depending on the volume of content, this might take some time to complete.

TFS 2018 Update 1.1 and TFS 2018 Update 3 include basic authentication for the communication between the TFS and Search service to make it more secure. Any user installing or upgrading to TFS 2018 Update 1.1 or TFS 2018 Update 3 will need to provide credentials as part of configuring Search feature (through Server or Search configuration wizard).

Manage Search in Azure DevOps Server and TFS

Search is managed by running PowerShell and SQL scripts. All of these scripts are available to download from [this GitHub repository](#). You may wish to download all of the scripts into a local folder on the server running the database for Azure DevOps Server using the **Download ZIP** option. The PowerShell scripts require the SQL script files, so ensure the **SqlScripts** folder and its contents is present, along with the PowerShell scripts.



NOTE

When executing scripts, ensure you run the appropriate script for your Azure DevOps Server or TFS version:

- [TFS 2017 RTM](#)
- [TFS 2017 Update 1](#)
- [TFS 2017 Update 2](#)
- [TFS 2017 Update 3](#)
- [TFS 2018 RTM](#)
- [TFS 2018 Update 1](#)
- [TFS 2018 Update 2](#)
- [Azure DevOps Server and TFS 2018 Update 3](#)

Check indexing status for TFS 2017 RTM

(For TFS 2017 Update 1 and later, and Azure DevOps Server, see the [next section](#))

To check the indexing status after Search is configured, or after the extension is installed for a collection:

1. Execute the **CheckIndexingStatus.ps1** script with administrative privileges. You will be prompted to enter:
 - The SQL server instance name where the TFS configuration database resides.
 - The name of the TFS collection database.
 - The name of the TFS configuration database.
 - The name of the collection.
 - The number of previous days to check indexing status.
2. Check the following outputs:
 - **Collection indexing was triggered successfully:** Indicates that indexing is in progress. If it is displayed, check the following outputs. If it is not displayed, go to step 3 below.
 - **Repositories Indexing Completed:** The repositories whose indexing has been completed and are now searchable.
 - **Repositories in File Discovery Phase:** The repositories where files are yet to be discovered. The files are indexed after this stage. Repositories in this state are not yet searchable. The number of files already discovered for indexing in each repository is shown, and this number should be increasing as more files are discovered.
 - **Repositories Indexing In Progress:** These repositories are partially indexed and should be searchable now, even if the results are only partial.
3. It takes some time for indexing to complete. Execute the **CheckIndexingStatus.ps1** script at intervals to check indexing progress.
4. If indexing is not occurring, or indexing is in progress but the number of files pending or the number of files discovered has not changed for some time, or no results are returned for a search, trigger indexing again by running the **TriggerCollectionIndexing.ps1** script in a PowerShell window with administrative permission.
5. If the problem persists, contact customer support at the address shown at the end of this topic.

Check indexing status for TFS 2017 Update 1 and later, and Azure DevOps Server

To check the indexing status after Search is configured, or after the extension is installed for a collection:

1. Execute the **ExtensionInstallIndexingStatus.ps1** script with administrative privileges. You will be prompted to enter:

- The SQL server instance name where the Azure DevOps Server or TFS configuration database resides.
 - The name of the Azure DevOps Server or TFS collection database.
 - The name of the Azure DevOps Server or TFS configuration database.
 - The name of the collection.
 - The number of previous days to check indexing status.
2. Check the following outputs:
- **Collection indexing was triggered successfully:** Indicates that indexing is in progress. If it is displayed, check the following outputs. If it is not displayed, go to step 3 below.
 - **Repositories completed indexing:** The number of repositories for which indexing has completed and are searchable.
 - **Status of repositories currently indexing:** A list of the names of all the repositories that are still being indexed and are partially searchable.
3. It takes some time for indexing to complete. Execute the **RecentIndexingActivity.ps1** script at intervals to check indexing progress. This script takes the same parameters as the **ExtensionInstallIndexingStatus.ps1** script.

- **Repositories completed fresh indexing:** The number of repositories for which indexing has completed within the specified time interval.
- **Count of repositories with fresh indexing in progress:** The number of repositories for which indexing has not yet completed. These repositories are still being indexed and are partially searchable.
- **Repositories completed continuous indexing:** The number of commits processed in the specified time interval. The number may not exactly match the total number of pushes to the repository because merges are committed as they are indexed.
- **Count of repositories with continuous indexing in progress:** The number of repositories for which the commits are still being processed. These repositories will show incomplete results until indexing is completed.
- **Count of indexing job failures:** The number of indexing jobs that failed. Repositories associated with these indexing jobs could potentially show incomplete results until subsequent indexing jobs for the same repositories have patched the failed indexing.

Pause indexing

To pause all indexing, execute the script **PauseSearchIndexing.ps1** with administrative privileges. Useful if you see spikes in CPU utilization after configuring Search.

You will be prompted to enter:

- The SQL server instance name where the Azure DevOps Server or TFS configuration database resides.
- The name of the Azure DevOps Server or TFS configuration database.

Resume indexing

If indexing was paused, execute the script **StartSearchIndexing.ps1** with administrative privileges, to start indexing again. You will be prompted to enter:

- The SQL server instance name where the Azure DevOps Server or TFS configuration database resides.
- The name of the Azure DevOps Server or TFS configuration database.

Re-index a repository or collection

To re-index a Git or TFVC repository, execute the appropriate version of the script **Re-IndexingRepository.ps1**

for your Azure DevOps Server or TFS version with administrative privileges. You will be prompted to enter:

- The SQL server instance name where the Azure DevOps Server or TFS configuration database resides.
- The name of the Azure DevOps Server or TFS collection database.
- The name of the Azure DevOps Server or TFS configuration database.
- The type of re-indexing to execute. This can be one of the values:
 - **Git_Repository**
 - **TFVC_Repository**
- The name of the collection.
- The name of the repository to re-index.

To re-index a **collection**, execute the script **TriggerCollectionIndexing.ps1** with administrative privileges. You will be prompted to enter:

- The SQL server instance name where the Azure DevOps Server or TFS configuration database resides.
- The name of the Azure DevOps Server or TFS collection database.
- The name of the Azure DevOps Server or TFS configuration database.
- The name of the collection.
- The entities to reindex. This can be one of the following values:
 - **All**
 - **Code**
 - **WorkItem**
 - **Wiki**

Re-indexing a collection can take from a few minutes to a few hours, depending on the size of the collection.

Also see [Troubleshoot Search](#).

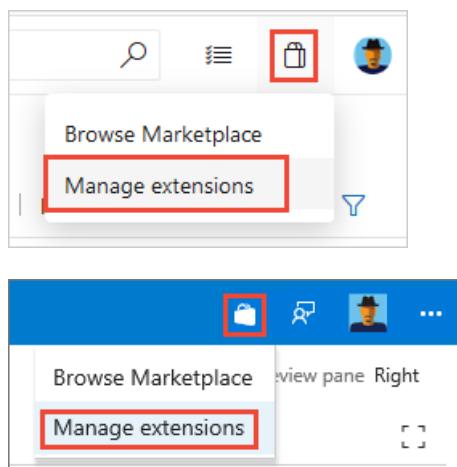
Uninstall Search from Azure DevOps Server or TFS

In cases such as a pre-production upgrade, production upgrade, new hardware migration, cloning, or other maintenance operation, the Server Configuration Wizard will unconfigure Search in a way that makes it easy to re-configure it after the server maintenance operation is complete.

However, there might be cases where you no longer want to use Search or you want to perform a new and clean install. This requires multiple steps, depending on whether Search is configured on the [same server](#) as Azure DevOps Server or TFS, or on a [separate server](#).

Unconfigure Search on the machine configured as your Azure DevOps Server

1. Uninstall the Search extension for each collection where it is installed. Do this by navigating to the **Manage Extensions** page of each collection in your Azure DevOps Server or TFS instance:



2. Remove the Search feature:

- Open the Azure DevOps Server or TFS Administration Console.
- In the left pane, select the name of the server.
- In the right pane, choose **Remove Feature**.
- In the Remove Feature dialog, select the **Search service** and choose **Remove**.

3. Remove the Elasticsearch service:

- Open **Command Prompt** as an administrator

- Change directory:

- For TFS 2017 RTM,

```
cd "C:\Program Files\Microsoft Team Foundation Server 15.0\Search\ES\elasticsearch-1.7.1-SNAPSHOT\bin"
```

- For TFS 2017 Update 1,

```
cd "C:\Program Files\Microsoft Team Foundation Server 15.0\Search\ES\elasticsearch-2.4.1\bin"
```

- For TFS 2018 Update 2 and above, and Azure DevOps Server,

```
cd "C:\Program Files\Microsoft Team Foundation Server 15.0\Search\ES\elasticsearch-5.4.1\bin"
```

- Remove the service:

- For TFS 2017, "service.bat remove"

- For TFS 2018 and Azure DevOps Server, "elasticsearch-service.bat remove"

4. Remove Search data:

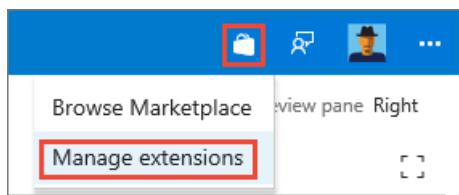
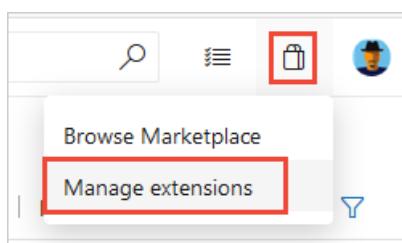
- Delete the contents of the location described by the environment variable `SEARCH_ES_INDEX_PATH`

5. Remove environment variables:

- Delete the environment variable `"SEARCH_ES_INDEX_PATH"`
- Delete the environment variable `"ES_HEAP_SIZE"` (this environment variable is obsolete for TFS 2018 Update 2 and later, and Azure DevOps Server).

Unconfigure Search when its configured on a separate server

1. Uninstall the Search extension (Code, Work Item, or Wiki) for each collection where it is installed. Do this by navigating to the **Manage Extensions** page of each collection in your Azure DevOps Server or TFS instance.



2. Remove the Search feature:

- Open the In the Remove Feature dialog, Administration Console.
- In the left pane, select the name of the Azure DevOps Server.

- In the right pane, choose **Remove Feature**.
 - In the Remove Feature dialog, select **Search service** and choose **Remove**.
3. Remove the Elasticsearch service and data
- Open **PowerShell** as an administrator
 - Go to the folder where **Configure Search.ps1** is installed along with the rest of the files required for a remote install of Search.
 - Run the script again with the remove option:
 - For TFS 2017 RTM, `"Configure Search.ps1 -RemoveTFSSearch"`
 - For TFS 2017 Update1 and above, and Azure DevOps Server, `"ConfigureTFSSearch.ps1 -remove"`

Limitations of Search in Azure DevOps Server and TFS

Search for Azure DevOps Server and TFS has the following limitation:

- If you perform a disaster recovery (DR) operation and move your server back to an earlier snapshot of your SQL database, you will need to [re-index all your collections](#).

Troubleshoot Search in Azure DevOps Server and TFS

- [Search is configured but the Search box is not displayed](#)
- [No search results are shown after installing or configuring Search](#)
- [How do I know if indexing was triggered for all the collections?](#)
- [Search stops working and no results are shown](#)
- [Search does not show the expected results](#)
- [Azure DevOps Server or TFS overall performance is affected](#)
- ["Unexpected error in Search service" message](#)

Search is configured but the Search box is not displayed

1. The search box is shown only in the context of a project page. Navigate to a project and check if the search box is displayed at the top right.
2. If the search box is not shown, verify that the extension is installed for the collection. If not, [install](#) or [configure](#) the extension.

No search results are shown after installing or configuring Search

1. Wait until you are sure sufficient time has elapsed after installing or configuring Search. It typically takes less than one hour for Search to index a collection, but it may take up to 12 hours depending on the size and number of code files, work items, or wiki pages.
2. If no results are shown after this period, [check indexing status](#).

How do I know if indexing was triggered for all the collections?

- [Check indexing status](#) separately for each collection.

Search stops working and no results are shown

Follow these steps. Replace "SearchServer" with the name of the server where Search is installed:

1. Access the URL `http://SearchServer:9200` from a web browser on a computer in the same domain as the server running Search.
 - If the status returned is `200 - OK`, go to step 2.

- If any other status is returned, contact support at the address shown at the end of this topic.
 - If you don't get a response, verify that the **elasticsearch-service-x64** service is running on the server where Search is configured. If the service is stopped, start it and access the Search server again.
If you still get no response, or a response other than `200 - OK`, contact support at the address shown at the end of this topic.
2. If the status is 200, access the URL `http://SearchServer:9200/_cat/health?v` from a web browser on a computer in the same domain as the server running Search.
- If the status column shows green/OK, and Search is still not working, contact support at the address shown at the end of this topic.
 - If the status column shows red/fault, look at the value in the **init** or **unassigned** columns. If these are greater than zero, wait for 30 minutes and then repeat this step. If the values are unchanged, go to step 3.
3. Access the URL `http://SearchServer:9200/_cat/shards?v` from a web browser on a computer in the same domain as the server running Search.
- Make a note of the values in the **Shard** column for the rows with a **state** value of **unassigned** and contact support at the address shown at the end of this topic.

Search does not show the expected results

1. If the files were added in the last few minutes, wait for ten minutes or so while they are indexed.
2. [Check indexing status](#) for the collection.
3. If the files are still not shown in the results, [re-index the repository or collection](#) where the files are located.

Azure DevOps Server or TFS overall performance is affected

1. [Pause all indexing](#) and see if performance recovers.
2. If performance does recover, consider locating Search on a separate server if you have not already done so.

Got feedback?



Report any problems on [Developer Community](#), or send feedback to vstssearch@microsoft.com.

Answers to common questions about Azure DevOps Search

8/1/2019 • 2 minutes to read • [Edit Online](#)

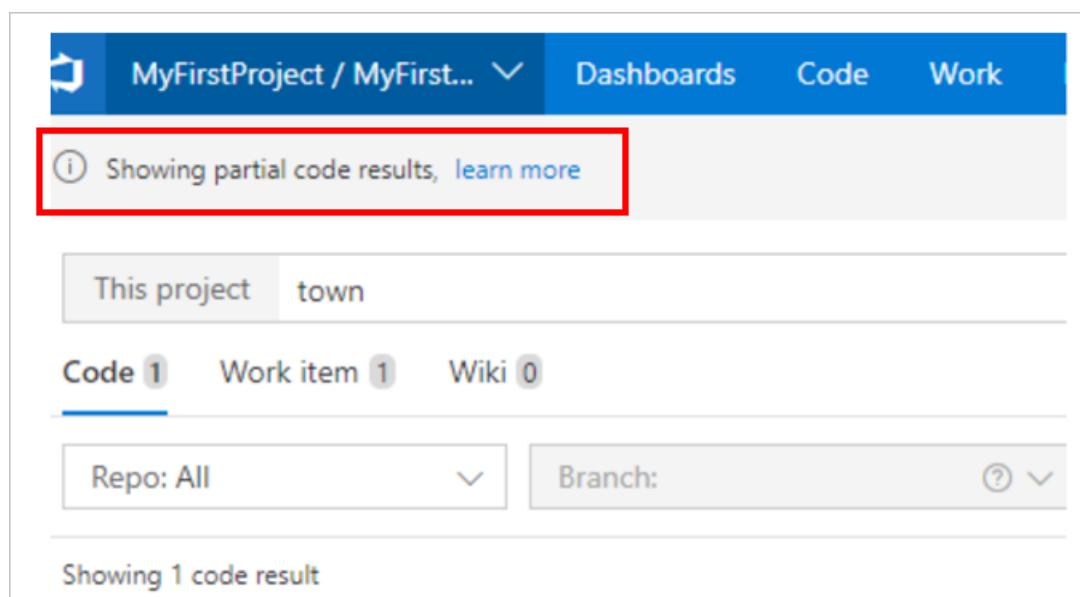
Azure DevOps Services | Azure DevOps Server 2019

This article provides answers to common questions about Azure DevOps Search.

Partial results in code search

Problem

I am seeing a **Showing partial code results** banner in code search.



The screenshot shows the Azure DevOps Code search interface. At the top, there's a navigation bar with 'MyFirstProject / MyFirst...' and dropdown menus for 'Dashboards', 'Code', and 'Work'. Below the navigation bar, a banner with a blue background and white text says '(i) Showing partial code results, learn more'. This banner is highlighted with a red rectangular box. Underneath the banner, there's a search bar with two tabs: 'This project' and 'town'. Below the search bar, there are three buttons: 'Code 1', 'Work item 1', and 'Wiki 0'. Further down, there are two dropdown menus: 'Repo: All' and 'Branch:'. At the bottom of the search area, it says 'Showing 1 code result'.

Explanation

You are likely to encounter this scenario when your code base has one or more large repositories (larger the repository, more the number of documents to search). So, when such repositories are searched, the request may take more time to process from all documents in the index and cause the search request to time out on the index. In such a case you may see partial search results along with **Showing partial code results** banner as shown above before the request times out.

Recommendation

You could try the following alternatives as applicable for your scenarios

- Try to scope your query by using filters to narrow down to a "repo" or a "path".
- See if your query itself could be narrowed down further to avoid scenarios that require matching too many terms while searching.

For example, while looking for methods like App_App1/ App_App2 etc., instead of searching for `a*` try searching for `app*` instead. (`a*` will match many more terms than `app*`).

Wildcard search

Problem

You may see different results while doing a wildcard search for the term `ge*` as compared to a wildcard search for the term `get*`. For example, in the image below you see `ge*` shows **7509** results.

This screenshot shows a search interface for a project. The search bar at the top contains the query `ge*`, which is highlighted with a red box. Below the search bar, there are navigation links: **Code 7K**, **Work item 50+**, **Wiki 50+**, and a link to **Search this organization**. Underneath these links are dropdown menus for **Repo: All**, **Branch:**, and **Path:**. A message box at the bottom left states **Showing 50 of 7509 code results**, also enclosed in a red box.

while `get*` shows **109,134** results.

This screenshot shows a search interface for a project. The search bar at the top contains the query `get*`, which is highlighted with a red box. Below the search bar, there are navigation links: **Code 109K**, **Work item 50+**, **Wiki 50+**, and a link to **Search this organization**. Underneath these links are dropdown menus for **Repo: All**, **Branch:**, and **Path:**. A message box at the bottom left states **Showing 50 of 109134 code results**, also enclosed in a red box.

Explanation

Let's understand how wildcard search works in the given scenario. Let's say you search for `app*`. In the backend, the wildcard `*` is expanded to match any character sequence after the term `app`. For example, `app*` might expand to `app, app0, app01, ..., apple`. This expansion takes place for the first 100 expanded terms only. Post the expansion, all the files associated with the 100 expanded terms are displayed on the search results page. In this case, there is a possibility that `application` may not be within the first 100 expanded terms therefore, you may not find files with the search term `application` in the search results. This is one of the reasons why you may see fewer search results for the term `ge*` as compared to `get*`.

Recommendation

This is to ensure that the search results remain performant and that you are able to find the most meaningful results as fast as possible. The expectation is that in case of wildcard search you can type more in the search bar to scope the results to a meaningful and actionable chunk.

Migrate & Import

10/2/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server | TFS

To migrate from an Azure DevOps on-premises deployment to Azure DevOps Services, review the articles provided in this migration guide.

For the full documentation set, go to the [Azure DevOps Documentation](#).

Migration and integration

- [Migrate data from TFS to Azure DevOps Services](#)
- [Migration options](#)
- [Import](#)
- [Process templates](#)
- [Post-import](#)
- [Troubleshooting](#)

Azure DevOps Services status & security

- [Service status](#)
- [Data protection](#)
- [Data location](#)
- [Credential storage](#)

DevOps: Continuous integration and continuous delivery

- [Add Continuous Security Validation to your CI/CD Pipeline](#)
- [Build and Deployment Automation Case Study](#)
- [Progressively expose your features using feature flags](#)
- [Progressively expose your releases using deployment rings](#)

Journey articles

Learn more about how we use DevOps and Agile practices at Microsoft, best practices research, lessons learned, and more.

- [Agile](#)
- [DevOps](#)
- [Git at Scale](#)

Resources

- [Git repos](#)
- [TFVC repos](#)

Migrate data from Azure DevOps Server to Azure DevOps Services

10/2/2019 • 5 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server](#) | [TFS](#)

NOTE

Visual Studio Team Services (VSTS) is now Azure DevOps Services.

On Tuesday, September 3rd support for Azure DevOps Server 2019 RTW imports will be deprecated. If you're planning to import from that version, please complete your import before September 3rd or upgrade to Azure DevOps Server 2019.0.1

Data migration tool for Azure DevOps

The data migration tool for Azure DevOps provides a high fidelity way to migrate collection databases from Azure DevOps Server to Azure DevOps Services. It's recommended that you download the [migration guide](#) if you're looking to use this service to import your collection(s). The guide serves as a walk through of the different steps involved in an import. Providing best practices, checklists, and helpful tips to make your import as easy as possible. The guide should be used in conjunction with the more technical documentation referenced below to successfully import to Azure DevOps Services.

Supported Azure DevOps Server versions for import

IMPORTANT

It can take up to 2-3 weeks after a new RTW version of Azure DevOps Server is released for import support to come online for that version. It's important to take this into consideration when choosing to upgrade shortly after a new RTW Azure DevOps Server release.

The data migration tool for Azure DevOps supports the two latest releases of Azure DevOps Server at a given time. Releases include updates and major releases. Currently the following versions of Azure DevOps Server are supported for import:

- Azure DevOps Server 2019.0.1
- Azure DevOps Server 2019 Update 1

NOTE

The data migration tool doesn't support imports from Azure DevOps Server release candidates (RC). If you're planning on importing your collection database to Azure DevOps Services using this service, it's important that you don't upgrade your production database to an RC release. If you do upgrade, then you will need to wait and upgrade to the release to web (RTW) version when it's available or restore a backup copy of your database from a previous Azure DevOps Server version to import.

Normal release cadence for new Azure DevOps Server versions is once every three-to-four months. Meaning that support for a given version of Azure DevOps Server for migration to Azure DevOps Services should last for anywhere between six-to-eight months. It's important to ensure that your planning accounts for this support

window to avoid having to suddenly upgrade to migrate.

Preview features

NOTE

If you're not including preview features when running the migration tool, then you will need to re-run the migration tool prepare to generate a new import.json to queue an import. You DO NOT need to include preview features when you regenerate your import.json.

If you had previously been including preview features then you DO NOT need to take any additional actions after Monday, April 23rd.

The following features can be included with your import, but are currently in a preview state.

- [Analytics](#) - Note this is only supported for Azure DevOps Server 2019 and later.

When queueing an import you can elect to include preview features with your import. If you do, data related to these features will be copied into your new organization along with all your other data. Should you choose to not include these features then their data will not be copied.

For a list of items not included with an import please see the [migration guide](#).

Data migration tool for Azure DevOps resources

In general you should use the [migration guide](#) when going through an import. When it's required the guide links back to the below documentation. These articles offer deeper technical knowledge on various import topics.

Import process

- [Validating a collection for import](#)
- [Preparing a collection for import](#)
- [Before import steps](#)
- [Run an Import](#)
- [Post import steps](#)

Troubleshooting

- [Troubleshooting validation errors](#)
- [Troubleshooting process errors](#)
- [Troubleshooting import errors](#)

Q & A

Q: Is there any risk of using the Hosting XML model becoming a problem in future updates of the service?

A: No, when it comes to service updates, Hosted XML organizations are treated the same as organizations using the Inheritance process model.

Q: Will my organization be stuck in Hosted XML forever?

A: You are using the Hosted XML process because the Inheritance process model does not contain all features yet. However, you can now [clone a hosted XML process to an Inheritance process](#).

Q: Will migrating from Hosted XML into Inheritance process model be a manual process?

A: No, the migration is automated. Simply follow the steps to [clone a hosted XML process to an Inheritance process](#).

Q: What happens in Hosted XML when Microsoft makes a change to a system process?

A: This is the same experience with Azure DevOps Server. If we make a change to a system process, it will not be applied to any of your Hosted XML processes. You won't have to update your processes if you don't want to. But if you do, you will need to make the changes in the XML definition files manually for each process.

Q: Is there a difference between a team project that was created manually versus one that was created from data import?

A. The features available to each team project are the same. The differences occur in how you modify the processes in your organization. When you create an organization, you will use the [Inheritance process model](#) to customize the work tracking experience. Team projects migrated via data import, however, will use the [Hosted XML process model](#) to customize the work tracking experience. As described above, these Hosted XML processes can be cloned to an Inheritance process model after import.

Q: If my organization is using Hosted XML, can I create new projects to use the Inheritance process model?

A: Yes. For data import organizations, Azure DevOps Services supports team projects that use Inheritance as well as Hosted XML process models. To learn more about the Inheritance process, see [Manage processes](#).

Q: Where can I find more information on Hosted XML and the Inheritance process model?

- [Inheritance Process Model](#)
- [Hosted XML](#)

Q: If I have feedback or additional questions is there somewhere I can reach out?

A: Yes, you can contact AzureDevOpsImport@microsoft.com. Please note that this alias is for general questions. If you need assistance with a failed import please contact Azure DevOps [customer support](#).

Videos

Migration options

10/2/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server | TFS

When you decide to make the move from Azure DevOps Server to Azure DevOps Services, you might start fresh with an empty organization. Often, however, you will have existing code, work items, and other assets that you want to move. There are many approaches to doing this which vary in both the fidelity of the data transfer and the complexity of the process.

Prior to migrating data, review the differences that exist between [Azure DevOps Server and Azure DevOps Services](#).

Option 1: Copy the most important assets manually

By far the easiest option for moving data into Azure DevOps Services is to manually copy your most important assets and start relatively fresh. This can be difficult when you are in the middle of a large project, but you can make it easier if you do some advance planning and schedule your move when it makes sense for your team.

For example, when the Azure DevOps team chose to move from Azure DevOps Server to Azure DevOps Services, we also decided to move from Team Foundation Version Control (TFVC) to Git. This required a fair bit of planning, but when we actually performed our migration, we created a new Git repo using the "tip" version of our TF VC sources, and left our history behind in Azure DevOps Server. We also moved our active work items, and left behind all our old bugs, completed user stories and tasks, and so on.

Here's the general process:

1. Identify the most important assets that you need to migrate - typically source code, work items, or both. Other assets in Azure DevOps Server - build pipelines, test plans, and so forth - are harder to manually migrate.
2. Identify a good time to make the transition.
3. Prepare your target organizations. Create the organizations and team projects that you need, provision users, and so on.
4. Migrate your data.
5. Consider making the source Azure DevOps Server deployments read-only.

Option 2: High fidelity database migration.

The Azure DevOps Server & Azure DevOps Services product team provides a high fidelity data migration tool. A downloadable Migration Guide is available at <https://aka.ms/AzureDevOpsImport>.



Because the data migration tool operates at a database level, it can provide a very high fidelity migration. If you want to move your existing Azure DevOps Server data into Azure DevOps Services, we strongly recommend using this option.

Option 3: Using public API-based tools for higher fidelity migration

If for some reason you cannot use the data migration tool but still want a higher fidelity migration than Option 1, you can choose from a variety of tools that use public APIs to move data. Generally these tools can provide a higher fidelity migration than a manual copy of "tip" data, but they are still relatively low fidelity. For example:

- None of them will preserve the dates of TF VC changesets.
- Many of them will not preserve the changed dates of work item revisions.
- None of them will migrate all Azure DevOps Server artifacts.

In general, we only recommend this approach if the extra fidelity beyond a manual copy is critical. If you decide to take this approach, you might consider hiring a consultant who has experience with one or more of the tools. You should definitely consider doing a test migration before doing your final migration.

Many organizations need a very high fidelity migration for only a subset of their work. New work could potentially start directly in Azure DevOps Services. Other work, with less stringent fidelity requirements, could be migrated using one of the other approaches. You will have to weigh the pros and cons of the various approaches against your motivations for moving into Azure DevOps Services and decide for yourself what is the right strategy.

Related articles

- [About Azure DevOps Services and Azure DevOps Server](#)
- [Pricing, Azure DevOps Services](#)
- [Pricing, Azure DevOps Server](#)

Import

10/2/2019 • 42 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server](#) | [TFS](#)

NOTE

Visual Studio Team Services (VSTS) is now Azure DevOps Services.

We recommend that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation, as needed.

Be sure you're on a [supported version](#) of Azure DevOps Server before continuing with the other import tasks.

With the release of Azure DevOps Server 2019 the TFS Database Import Service has been rebranded to become data migration tool for Azure DevOps. This includes Tfsmigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

This page walks through how to perform all of the necessary preparation work required to get an import to Azure DevOps Services ready to run. If you encounter errors during the process be sure to review the [troubleshooting](#).

Validating a collection

Now that you've confirmed you're on the latest version of Azure DevOps Server the next step is to validate each collection you wish to migrate to Azure DevOps Services. Validate will examine a variety of aspects in your collection, including, but not limited to: size, collation, identity, and processes. Running a validation is done through the data migration tool. To start, take a copy of the [data migration tool](#) and copy it onto one of your Azure DevOps Server Application Tiers (AT). Once there you can unzip it. The tool can also be run from a different machine without Azure DevOps Server installed as long as the PC can connect to the Azure DevOps Server instance's configuration database - example below.

To get started, open a command prompt on the server and CD to the path where you have the data migration tool placed. Once there it's recommended that you take a second to review the help text provided with the tool. Run the following command to see the top level help and guidance:

```
Migrator /help
```

For this step, we'll be focusing on the validate command. To see the help text for that command simply run:

```
Migrator validate /help
```

Since this is our first time validating a collection we'll keep it simple. Your command should have the following structure:

```
Migrator validate /collection:{collection URL}
```

For example, to run against the default collection the command would look like:

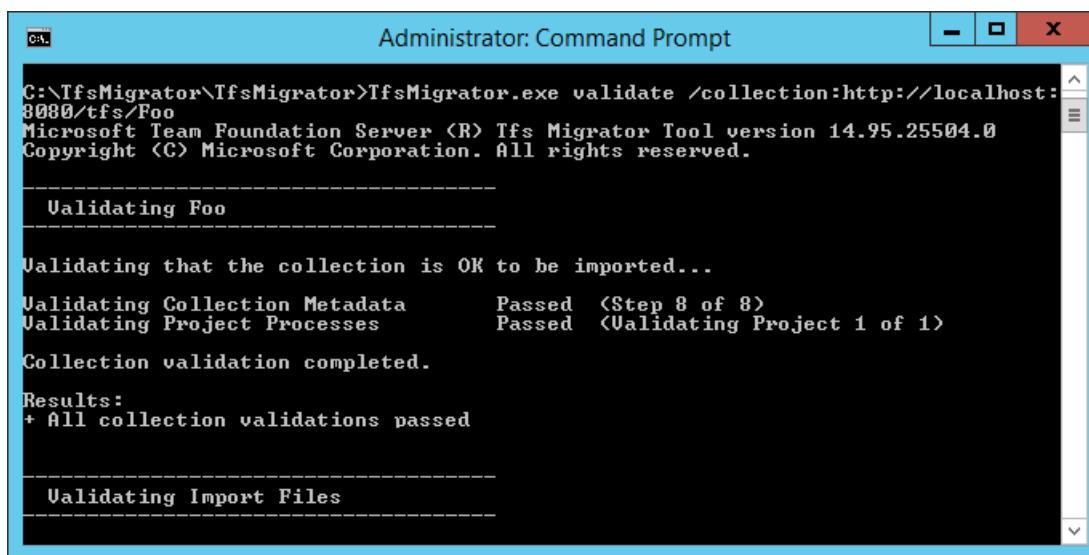
```
Migrator validate /collection:http://localhost:8080/DefaultCollection
```

Running it from a machine other than the Azure DevOps Server requires the /connectionString parameter. The connection string parameter is a pointer to your Azure DevOps Server configuration database. As an example, if the validate command was being run by the Fabrikam corporation the command would look like:

```
Migrator validate /collection:http://fabrikam:8080/DefaultCollection  
/tenantDomainName:fabrikam.OnMicrosoft.com /connectionString:"Data Source=fabrikam;Initial  
Catalog=Configuration;Integrated Security=True"
```

It's important to note that the data migration tool **DOES NOT** edit any data or structures in the collection. It only reads the collection to identify issues.

Once the validation is complete you'll be left with a set of log files and a set of results printed to the command prompt screen.



The screenshot shows an Administrator Command Prompt window with the title "Administrator: Command Prompt". The window displays the output of the TfsMigrator validate command. The output includes the command used, copyright information, validation steps (Validating Foo, Validating Collection Metadata, Validating Project Processes), and a summary stating "All collection validations passed".

```
C:\TfsMigrator\TfsMigrator>TfsMigrator.exe validate /collection:http://localhost:8080/tfs/Foo  
Microsoft Team Foundation Server (R) Tfs Migrator Tool version 14.95.25504.0  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
-----  
Validating Foo  
  
Validating that the collection is OK to be imported...  
Validating Collection Metadata Passed <Step 8 of 8>  
Validating Project Processes Passed <Validating Project 1 of 1>  
Collection validation completed.  
Results:  
+ All collection validations passed  
  
-----  
Validating Import Files
```

If all of the validations pass, you are ready to move onto the next step of the import process. If the data migration tool flagged any errors, they will need to be corrected before moving on. See [troubleshooting](#) for guidance on correcting validation errors.

When you open up the log directory you will notice that there are several logging files.

Collection.log	7/19/2016 2:29 PM	Text Document
ProjectProcessesMap.log	7/19/2016 2:29 PM	Text Document
TfsMigrator.log	7/19/2016 2:29 PM	Text Document
TryMatchOobProcesses.log	7/19/2016 2:29 PM	Text Document

The log titled `DataMigrationTool.log` is going to be the main log which contains details on everything that was run. To make it easier to narrow down on specific areas, a log is generated for each major validation operation. For example, if TfsMigrator had reported an error in the "Validating Project Processes" step, then one can simply open the `ProjectProcessMap.log` file to see everything that was run for that step instead of having to scroll through the overall log. The `TryMatchOobProcesses.log` should only be reviewed if you're trying to import your project processes to use the [inherited model](#). If you don't want to use the new inherited model then the errors in this file will not prevent you from doing an import to Azure DevOps Services and can be ignored.

Generating import files

By this point you will have run the data migration tool *validate* against the collection and it is returning "All collection validations passed". Before you start taking the collection offline to migrate, there is some more

preparation that needs to be completed - generating the import files. Upon running the prepare step, you will generate two import files: `IdentityMapLog.csv` which outlines your identity map between Active Directory (AD) and Azure Active Directory (Azure AD), and `import.json` which requires you to fill out the import specification you want to use to kick off your migration.

Prepare command

The prepare command assists with generating the required import files. Essentially, this command scans the collection to find a list of all users to populate the identity map log, `IdentityMapLog.csv`, and then tries to connect to Azure AD to find each identity's match. Your company will need to employ the Azure Active Directory Connect [tool](#) (formerly known as the Directory Synchronization tool, Directory Sync tool, or the DirSync.exe tool). If directory synchronization is setup, the data migration tool should be able to find the matching identities and mark them as Active. If it doesn't find a match, the identity will be marked Historical in the identity map log and you will need to investigate why the user wasn't included in your directory sync. The Import specification file, `import.json`, should be filled out prior to importing.

Unlike the validate command, prepare **DOES** require an internet connection as it needs to reach out to Azure AD in order to populate the identity map log file. If your Azure DevOps Server instance doesn't have internet access, you'll need to run the tool from a different PC that does. As long as you can find a PC that has an intranet connection to your Azure DevOps Server instance and an internet connection then you can run this command. Run the following command to see the guidance for the prepare command:

```
Migrator prepare /help
```

Included in the help documentation are instructions and examples for running Migrator from the Azure DevOps Server instance itself and a remote PC. If you're running the command from one of the Azure DevOps Server instance's Application Tiers (ATs) then your command should have the following structure:

```
Migrator prepare /collection:{collection URL} /tenantDomainName:{name} /region:{region}
```

```
Migrator prepare /collection:{collection URL} /tenantDomainName:{name} /region:{region}  
/connectionString:"Data Source={sqlserver};Initial Catalog=Configuration;Integrated Security=True"
```

The connection string parameter is a pointer to your Azure DevOps Server instance configuration database. As an example, if the prepare command was being run by the Fabrikam corporation the command would look like:

```
Migrator prepare /collection:http://fabrikam:8080/DefaultCollection /tenantDomainName:fabrikam.OnMicrosoft.com  
/region:{region} /connectionString:"Data Source=fabrikam;Initial Catalog=Configuration;Integrated  
Security=True"
```

Upon executing this command, the data migration tool will run a complete validate to ensure that nothing has changed with your collection since the last full validate. If any new issues are detected, then the import files will not be generated. Shortly after the command has started running, an Azure AD login window will appear. You will need to sign in with an identity that belongs to the tenant domain specified in the command. It's important to make sure that the Azure AD tenant specified is the one you want your future organization to be backed with. For our Fabrikam example the user would enter something similar to what's shown in the below image.

IMPORTANT

Do NOT use a test Azure AD tenant for a test import and your production Azure AD tenant for the production run. Using a test Azure AD tenant can result in identity import issues when you begin your production run with your organization's production Azure AD tenant.

Sign in to your account X

Microsoft Azure

Work or school, or personal Microsoft account

Sign in Back

[Can't access your account?](#)

© 2016 Microsoft  Microsoft
[Terms of use](#) [Privacy & Cookies](#)

A successful run of the data migration tool prepare will result in a set of logs and two import files.

After opening the log directory noted in the data migration tool's output you will notice that there are two files and a Logs folder. `IdentityMapLog.csv` contains the generated mapping of AD to Azure AD identities. `import.json` is the import specification file which needs to be filled out. It's recommended that you take time to fill out the import specification file, `import.json`, and review the identity map log file, `IdentityMapLog.csv`, for completeness before kicking off an import.

Import specification file

The import specification, `import.json`, is a JSON file which provides import settings. It includes information such as the desired organization name, storage account information, etc. Most of fields are auto-populated, some fields require user input prior to attempting an import.

```

1  [
2    "Source": {
3      "Location": "<Provide the SASKey to the Azure storage container with the collection and import files.>",
4      "Files": {
5        "Dacpac": "Tfs_DefaultCollection.dacpac"
6      }
7    },
8    "Target": {
9      "Name": "<Provide a name for the account that will be created during the import.>"
10     },
11    "Properties": {
12      "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"
13    },
14    "ValidationData": {
15      "TfsMigratorVersion": "16.255.65000.0",
16      "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17      "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18      "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19      "CommandExecutionCount": 0,
20      "CommandExecutionTime": 0.0,
21      "TfsVersion": "Dev15.M117",
22      "DatabaseTotalSize": 181,
23      "DatabaseBlobSize": 0,
24      "DatabaseTableSize": 181,
25      "DatabaseLargestTableSize": 8,
26      "ActiveUserCount": 8,
27      "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28      "Region": "CUS",
29      "ValidationChecksumVersion": 1,
30      "ValidationChecksum":
31        "66S16G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
32    },
33    "Identities": [
34      "S-1-5-21-1374400868-3601225936-2087002269-500",
35      "S-1-5-21-2127521184-1604012920-1887927527-11008431",
36      "S-1-5-21-2127521184-1604012920-1887927527-15795496"
37    ]
]

```

0 0 1 json | import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON

Here is the breakdown of the fields and what action needs to be taken:

FIELD	EXPLANATION	ACTION
Source	Information detailing location and names of source data files used for import.	None - Review information for subfield actions below.
ValidationData	Contains information needed to help drive your import experience	The 'ValidationData' section is generated by the data migration tool. It contains information needed to help drive your import experience, so it's important that you don't edit the values inside of this section or your import could fail to start.
Files	Name of the files containing import data.	None - Review information for subfield actions below.
Target	Properties describing the new organization to import into.	None - Review information for subfield actions below.

FIELD	EXPLANATION	ACTION
Name	Desired name for the organization that will be created during the import.	Select a name. This name can be quickly changed later after the import has completed. Note – do NOT create an organization with this name before running the import. The organization will be created as part of the import process.
Location	SAS Key to the Azure storage account hosting the DACPAC.	None – This will be covered in a later step.
Dacpac	A file that packages up your collection database that is used to bring the data in during import.	None - In a later step you'll generate this file using your collection and will upload it to an Azure storage account. It will need to be updated based on the name you use when generating the DACPAC later in this process.
ImportType	The type of import that you want to run.	None - In a later step you will select the type of import to run.

After following the above instructions, you should have a file that looks somewhat like the below.

```
import.json - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
import.json x
1  {
2      "Source": {
3          "Location": "<Provide the SASKey to the Azure storage container with the collection and import files.>",
4          "Files": {
5              "Dacpac": "Tfs_DefaultCollection.dacpac"
6          }
7      },
8      "Target": {
9          "Name": "fabrikam-import"
10     },
11     "Properties": {
12         "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"
13     },
14     "ValidationData": {
15         "TfsMigratorVersion": "16.255.65000.0",
16         "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17         "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18         "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19         "CommandExecutionCount": 0,
20         "CommandExecutionTime": 0.0,
21         "TfsVersion": "Dev15.M117",
22         "DatabaseTotalSize": 181,
23         "DatabaseBlobSize": 0,
24         "DatabaseTableSize": 181,
25         "DatabaseLargestTableSize": 8,
26         "ActiveUserCount": 8,
27         "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28         "Region": "CUS",
29         "ValidationChecksumVersion": 1,
30         "ValidationChecksum":
31             "66S16G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
32     },
33     "Identities": [
34         "S-1-5-21-1374400868-3601225936-2087002269-500",
35         "S-1-5-21-2127521184-1604012920-1887927527-11008431",
36         "S-1-5-21-2127521184-1604012920-1887927527-15795496"
37     ]
}
Ln 1, Col 1  Spaces: 2  UTF-8  CRLF  JSON  🌐
```

In this case, the user planning the Fabrikam import added the organization name "Fabrikam-Import" and selected the Central US region in the Target object. Other values were left as is to be modified just before taking the collection offline for the migration.

NOTE

Dry run imports will have a '-dryrun' automatically appended to the end of the organization's name. This can be changed post import.

Supported Azure Regions for Import

Azure DevOps Services is available in several Azure [regions](#). However, not all Azure regions that Azure DevOps Services is present in are supported for import. The following table details the Azure regions that can be selected for import. Also included is the value which needs to be placed in the import specification file to target that region for import.

GEOGRAPHIC REGION	AZURE REGION	IMPORT SPECIFICATION VALUE
United States	Central United States	CUS

GEOGRAPHIC REGION	AZURE REGION	IMPORT SPECIFICATION VALUE
Europe	Western Europe	WEU
United Kingdom	United Kingdom South	UKS
Australia	Australia East	EAU
South America	Brazil South	SBR
Asia Pacific	South India	MA
Asia Pacific	East Asia (Hong Kong)	EA
Canada	Central Canada	CC

Identity map log

Arguably the identity map log is of equal importance to the actual data that you will be migrating to Azure DevOps Services. When reviewing the file it's important to understand how identity import operates and what the potential results could entail. When importing an identity, they could either end up becoming active or historical. The difference between active and historical identities is that active identities can log into Azure DevOps Services whereas historical identities cannot. It's important to note that once imported as a historical identity, there is no way to move that identity to become active.

Active identities

Active identities refer to identities that will be users in Azure DevOps Services post-import. On Azure DevOps Services, these identities will be licensed and show up as a user in the organization after migration. These identities are marked as 'active' in the "Expected Import Status" column in the identity map log file.

Historical identities

These are identities that are mapped as 'historical' in the "Expected Import Status" column in the identity map log file. Also, identities that have no line entry present in the file will also become historical. An example of an identity with no entry would be an employee that no longer works at a company. Historical Identities do **NOT** have access to an organization after migration, do **NOT** have a licenses, and do **NOT** show up as a user in the organization. All that is persisted is the notion of that identity's name in the organization. This way their history can be searched at a later date. It's recommended that historical identities be used for users that are no longer at the company or won't ever be needing access to the organization. Identities imported historically **CANNOT** be migrated later to become active identities.

Understanding an Identity Map Log

After opening the identity map log file, you will be presented with something similar to the below example.

AD: User(TFS)	AD: Security Identifier	AAD: Expected Import User(VSTS)	Expected Import Status	Validation Date
FABRIKAM\Jamal Hartnett	S-1-5-21-983578539-230207283-3682864982-500	No Match Found (Check AAD Sync)	Historical	2017-10-31T21:15:44Z
FABRIKAM\Mateo Escobedo	S-1-5-21-4100298327-4227319834-4140607669-500	No Match Found (Check AAD Sync)	Historical	2017-10-31T21:15:44Z
FABRIKAM\Helena Petersen	S-1-5-21-124525095-708259637-1543119021-1419599	helena.petersen@fabrikam.com	Active	2017-10-31T21:15:44Z
FABRIKAM\Raisa Pokrovskaya	S-1-5-21-2127521184-1604012920-1887927527-406986	raisa.pokrovskaya@fabrikam.com	Active	2017-10-31T21:15:44Z

The table below explains what each column is used for.

NOTE

Users marked as "No Match Found (Check Azure AD Sync)" who you wanted to be added as full organization members will need to be investigated with your Azure AD admin to see why they aren't part of your Azure AD Connect sync.

COLUMN	EXPLANATION
AD - User (Azure DevOps Server)	Friendly display name used by the identity in Azure DevOps Server. Makes it easier to identify which user the line in the map is referencing.
AD - Security Identifier	The unique identifier for the on-prem AD identity in Azure DevOps Server. This column is used to identify users in the collection.
Azure AD - Expected Import User (Azure DevOps Services)	Either the expected sign in address of the matched soon to be active user or "No Match Found (Check Azure AD Sync)" indicating that the identity was not found in during AAd sync and will be imported as historical.
Expected Import Status	The expected user import status, either "Active" if there was a match between your AD and Azure AD or "Historical" if we could not match the AD identity in your Azure AD.
Validation Date	Last time the identity map log was validated.

Reading through the file you will notice the Expected Import Status column has either 'Active' or 'Historical'. Active indicates that it's expected that the identity on this row will map correctly on import and will become active. Historical will become historical identities on import. It's important that you review the generated mapping file for completeness and correctness.

IMPORTANT

Your import will fail if major changes occur to your Azure AD Connect SID sync between import attempts. New users can be added between dry runs, and corrections to ensure previously imported historical identities become active are also OK. However, changing an existing user that was previously imported as active is not supported at this time. Doing so will cause your import to fail. For example, completing a dry run import, deleting an identity from your Azure AD that was imported actively, recreating a new user in Azure AD for that same identity, and attempt another import. In this case an active identity import will be attempted between the AD and newly created Azure AD identity, but it will cause an import failure as this isn't supported.

Start by reviewing the correctly matched identities. Are all of the expected identities present? Are the users mapped to the correct Azure AD identity? If any values are incorrectly mapped or need to be changed then you'll need to contact your Azure AD administrator to check whether the on-premises Active Directory (AD) identity is part of the sync to Azure AD and has setup correctly. Check the [documentation](#) on setting a sync between your on-premises Active Directory (AD) and Azure AD.

Next, review the identities that are labeled as 'Historical'. This implies that a matching Azure AD identity couldn't be found. This could be for one of four reasons.

1. The identity hasn't been setup for sync between on-premises Active Directory (AD) and Azure AD.
2. The identity hasn't been populated in your Azure AD yet; new employee scenario.
3. The identity simply doesn't exist in your Azure AD.
4. The user that owned that identity no longer works at the company.

In the first three cases the desired on-premises Active Directory (AD) identity will need to be set up for sync with Azure AD. Check the [documentation](#) on setting a sync between your on-premises Active Directory (AD) and Azure AD. It's required that Azure AD Connect be setup and run for identities to be imported as active in Azure DevOps Services. The final case can generally be ignored as employees no longer at your company should be imported historically.

Historical Identities (Small Teams)

The identity import strategy proposed in this section should only be considered by small teams.

In cases where the Azure AD Connect hasn't been configured, you will notice that all users in the identity map log file will be marked as 'Historical'. Running an import this way will result in all users getting imported [historically](#). It's strongly recommended that you configure [Azure AD Connect](#) to ensure that your users are imported as active.

Running an import with all historical identities has consequences which need to be considered carefully. It should only be considered by teams with a small number of users were the cost of setting up an Azure AD Connect is deemed too high.

To import with all historical identities, simply follow the steps outlined in later sections. When queuing an import, the identity that is used to queue the import will be bootstrapped into the organization as the organization owner. All other users will be imported historically. The organization owner will then be able to [add users](#) back in using their Azure AD identity. Users added will be treated as new users. They will **NOT** own any of their history and there is no way to re-parent this history to the Azure AD identity. However, users can still lookup their pre-import history by searching for {domain}{AD username}.

The data migration tool will warn if it detects the complete historical identities scenario. If you decide to go down this migration path you will need to consent in the tool to the limitations.

Visual Studio Subscriptions

The data migration tool is unable to detect Visual Studio subscriptions (formerly known as MSDN benefits) when generating the identity map log file. Instead, it's recommended that you leverage the auto license upgrade feature post import. As long as a user's work account is [linked](#) correctly, Azure DevOps Services will automatically apply their Visual Studio subscription benefits on their first login post import. You're never charged for licenses assigned during import, so this can be safely handled post import.

You don't need to repeat a dry run import if users don't automatically get upgraded to use their Visual Studio Subscription in Azure DevOps Services. Visual Studio Subscription linking is something that happens outside of the scope of an import. As long as the work account gets linked correctly before or after the import then the user will automatically have their license upgraded on the next sign in. Once they've been upgraded successfully, next time you import the user will be upgraded automatically on the first sign in to the organization.

Getting Ready to Import

By this point you will have everything ready to execute on your import. You will need to schedule downtime with your team to take the collection offline for the migration. Once you have an agreed upon a time to run the import you need to get all of the required assets you have generated and a copy of the database uploaded to Azure. This process has five steps:

1. Take the collection offline and detach it.
2. Generate a DACPAC from the collection you're going to import.
3. Upload the DACPAC and import files to an Azure storage account.
4. Generate a SAS Key to that storage account.
5. Fill out the last fields in the import specification.

NOTE

We **strongly** recommend that your organization complete a dry run import before performing a production import. Dry runs allow you to validate that the import process works for your collection and that there are no unique data shapes present which might cause a production import failure.

Detaching your Collection

Detaching the collection is a crucial step in the import processes. Identity data for the collection resides in the Azure DevOps Server server's configuration database while the collection is attached and online. When a collection is detached from the Azure DevOps Server instance it will take a copy of that identity data and package it up with the collection for transport. Without this data the identity portion of the import **CANNOT** be executed. It's recommended that the collection stay detached until the import has been completed, as there isn't a way to import the changes which occurred during the import.

If you're running a dry run (test) import, it's recommended to reattach your collection after backing it up for import since you won't be concerned about having the latest data for this type of import. You could also choose to employ an [offline detach](#) for dry runs to avoid offline time all together. It's important to weigh the cost involved with going the zero downtime route for a dry run. It requires taking backups of the collection and configuration database, restoring them on a SQL instance, and then creating a detached backup. A cost analysis could prove that taking just a few hours of downtime to directly take the detached backup is better in the long run.

Generating a DACPAC

IMPORTANT

Before proceeding, ensure that your collection was [detached](#) prior to generating a DACPAC.

NOTE

If the data migration tool didn't warn that your collection was too big, use the DACPAC method outlined below. Otherwise see the section on importing large collections at <https://aka.ms/AzureDevOpsImportLargeCollection>.

Data-tier Application Component Packages ([DACPAC](#)) is a feature in SQL server that allows database changes to be packaged into a single file and deployed to other instances of SQL. It can also be restored directly to Azure DevOps Services and is therefore utilized as the packaging method for getting your collection's data in the cloud. You're going to use the SqlPackage.exe tool to generate the DACPAC. This tool is included as part of the [SQL Server Data Tools](#).

There are multiple versions of SqlPackage.exe installed with SQL Server Data Tools, located under folders with names such as 120, 130, and 140. When using SqlPackage.exe it is important to use the right version to prepare the DACPAC.

- TFS 2018 imports need to use SqlPackage.exe from the 140 folder or higher.

If you installed SQL Server Data Tools (SSDT) for Visual Studio, you can find SqlPackage.exe in one of the following locations.

- If you installed SSDT and integrated it with an existing installation of Visual Studio, SqlPackage.exe is located in a folder similar to:
`C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\Extensions\Microsoft\SQLDB\DAC\130\`
- If you installed SSDT and did a stand-alone installation, SqlPackage.exe is located in a folder similar to:
`C:\Program Files (x86)\Microsoft Visual Studio\2017\SQL\Common7\IDE\Extensions\Microsoft\SQLDB\DAC\130\`
- If you already have an installation of SQL Server, SqlPackage.exe may already be present, and located in a folder similar to: `%PROGRAMFILES%\Microsoft SQL Server\130\DAC\bin\`

Both of the versions of SSDT that you can download from [SQL Server Data Tools](#) include both the 130 and 140 folders with their respective versions of SqlPackage.exe.

When generating a DACPAC there are two considerations that you'll want to keep in mind, the disk that the DACPAC will be saved on and the space on disk for the machine performing the DACPAC generation. Before

generating a DACPAC you'll want to ensure that you have enough space on disk to complete the operation. While creating the package, SqlPackage.exe temporarily stores data from your collection in the temp directory on the C: drive of the machine you initiate the packaging request from. Some users might find that their C: drive is too small to support creating a DACPAC. Estimating the amount of space you'll need can be found by looking for the largest table in your collection database. As DACPACs are created one table at a time. The maximum space requirement to run the generation will be roughly equivalent to the size of the largest table in the collection's database. You will also need to take into account the size of the collection database as reported in DataMigrationTool.log file from a validation run, if you choose to save the generated DACPAC on the C: drive.

DataMigrationTool.log provides a list of the largest tables in the collection each time the validate command is run. See the example below for a sample output showing table sizes for a collection. Compare the size of the largest table with the free space on the drive hosting your temporary directory.

	Size in MB
[Info @08:23:59.539] Table name	
[Info @08:23:59.539] dbo.tbl_Content	38984
[Info @08:23:59.539] dbo.tbl_LocalVersion	1935
[Info @08:23:59.539] dbo.tbl_Version	238
[Info @08:23:59.539] dbo.tbl_FileReference	85
[Info @08:23:59.539] dbo.Rules	68
[Info @08:23:59.539] dbo.tbl_MetaData	61

Ensure that the drive hosting your temporary directory has at least that much free space. If it doesn't then you'll need to redirect the temp directory by setting an environment variable.

```
SET TEMP={location on disk}
```

Another consideration is where the DACPAC data is saved. Pointing the save location to a far off remote drive could result in much longer generation times. It's recommended that if a fast drive, such as an SSD, is available locally that you target that drive as the DACPAC's save location. Otherwise, it's always faster to use a disk that's on the machine where the collection database is residing over a remote drive.

Now that you've identified the target location for the DACPAC and ensured that you'll have enough space, it's time to generate the DACPAC file. Open a command prompt and navigate to the location where SqlPackage.exe is located. Taking the command example below, replace the required values and generate the DACPAC

```
SqlPackage.exe /sourceconnectionstring:"Data Source={database server name};Initial Catalog={Database Name};Integrated Security=True" /targetFile:{Location & File name} /action:extract /p:ExtractAllTableData=true /p:IgnoreUserLoginMappings=true /p:IgnorePermissions=true /p:Storage=Memory
```

- **Data Source** - SQL Server instance hosting your Azure DevOps Server collection database.
- **Initial Catalog** - Name of the collection database.
- **targetFile** - Location on disk + name of DACPAC file.

Below is an example of the DACPAC generation command that is running on the Azure DevOps Server data tier itself:

```
SqlPackage.exe /sourceconnectionstring:"Data Source=localhost;Initial Catalog=Foo;Integrated Security=True" /targetFile:C:\DACPAC\Foo.dacpac /action:extract /p:ExtractAllTableData=true /p:IgnoreUserLoginMappings=true /p:IgnorePermissions=true /p:Storage=Memory
```

The output of the command will be a DACPAC that is generated from the collection database Foo called Foo.dacpac.

Importing Large Collections

NOTE

If the data migration tool warns that you can't use the DACPAC method then you will have to import using the SQL Azure VM method outlined below. If the data migration tool didn't warn that your collection was too big, use the DACPAC method outlined above.

DACPACs offer a fast and relatively simplistic method for moving collections into Azure DevOps Services. However, once a collection database crosses a certain size threshold the benefits of using a DACPAC start to diminish. For databases that the data migration tool warns are too big, a different data packaging approach is required to migrate to Azure DevOps Services. If you're unsure if your collection is over the size threshold then you should run a data migration tool validate on the collection. The validation will let you know if you need to use the SQL Azure VM method for import or not.

Before going any further, it's always recommended to see if [old data can be cleaned up](#). Over time collections can build up very large volumes of data. This is a natural part of the DevOps process. However, some of this data might no longer be relevant and doesn't need to be kept around. Some common examples are older workspaces and build results. Cleaning older, no longer relevant artifacts might remove a lot more space than one would expect. It could be the difference between using the DACPAC import method or having to use a SQL Azure VM. It's important to note that once you deleted older data that it **CANNOT** be recovered without restoring an older backup of the collection.

If you are under the DACPAC threshold, follow the instructions to [generate a DACPAC](#) for import. If you're still unable to get the database under the DACPAC threshold then you will need to setup a SQL Azure VM to import to Azure DevOps Services. We'll walk through how to accomplish this end-to-end. At a high-level the steps covered include:

1. Setting up a SQL Azure VM
2. Optionally, we recommend restricting access to just Azure DevOps Services IPs
3. Restoring your database on the VM
4. Creating an identity to connect to the collection database
5. Configuring your import specification file to use a SQL connection string

Creating the SQL Azure VM

Setting up a SQL Azure VM can be done from the Azure portal with just a few clicks. Azure has a [tutorial](#) on how to setup and configure a SQL Azure VM.

Azure DevOps Services is available in several Azure [regions](#) across the globe. When importing to these regions it's critical that you place your data in the correct region to ensure that the import can start correctly. Setting up your SQL Azure VM in a location other than the ones recommended below will result in the import failing to start.

Use the table below to decide where you should create your SQL Azure VM if you're using this method to import. Creating your VM in a region outside of the list below is not supported for running an import.

DESIRED IMPORT REGION	SQL AZURE VM REGION
Central United States	Central United States
Western Europe	Western Europe
Australia East	Australia East
Brazil South	Brazil South

DESIRED IMPORT REGION	SQL AZURE VM REGION
South India	South India
Central Canada	Central Canada
East Asia (Hong Kong)	East Asia (Hong Kong)
UK South	UK South

While Azure DevOps Services is available in multiple regions in the United States, only the Central United States region is accepting new organizations. Customers will not be able to import their data into other United States Azure regions at this time.

NOTE

DACPAC customers should consult the region table in the [uploading DACPAC and import files section](#). The above guidelines are for SQL Azure VMs only.

Below are some additional recommended configurations for your SQL Azure VM.

1. It's recommended that D Series VMs be used as they're optimized for database operations.
2. Ensure that the D Series VM has at least 28GBs of ram. Azure D12 V2 VM sizes are recommended for imports.
3. [Configure](#) the SQL temporary database to use a drive other than the C drive. Ideally this drive should have ample free space; at least equivalent to your database's [largest table](#).
4. If your source database is still over 1TB after [reducing the size](#) then you will need to [attach](#) additional 1TB disks and combine them into a single partition to restore your database on the VM.
5. Collection databases over 1TB in size should consider using Solid State Drives (SSDs) for both the temporary database and collection database.

Azure DevOps Services IPs

It's highly recommended that you restrict access to your VM to only IPs from Azure DevOps Services. This can be accomplished by allowing connections only from the set of Azure DevOps Services IPs that are involved in the collection database import process. The IPs that need to be granted access to your collection database will depend on what region you're importing into. The tables below will help you identify the correct IPs. The only port that is required to be opened to connections is the standard SQL connection port 1433.

First, no matter what Azure DevOps Services region you import into the following IP must be granted access to your collection database.

SERVICE	IP
Azure DevOps Services Identity Service	168.62.105.45, 40.81.42.115

Next you will need to grant access to the Regional Identity Service. You only need to grant an exception for the data migration tool instance in the region that you're importing into.

SERVICE	IP
Regional Identity Service - Central United States	23.99.230.232, 104.43.253.175, 40.122.66.150, 40.122.117.178, 23.99.212.58, 23.99.214.58, 20.37.139.247, 20.37.138.167, 20.37.138.122

SERVICE	IP
Regional Identity Service - West Europe	52.232.119.33, 104.46.44.17, 40.114.142.95, 51.144.180.30, 51.145.130.146, 40.113.97.58, 52.232.113.92, 40.74.51.167
Regional Identity Service - Australia East	13.70.121.123, 52.187.228.246
Regional Identity Service - Brazil South	N/A
Regional Identity Service - India South	104.211.226.91, 104.211.207.31, 40.81.75.134
Regional Identity Service - Canada Central	13.88.230.114, 40.85.244.98, 40.82.185.245
Regional Identity Service - East Asia (Hong Kong)	23.98.36.60, 40.83.79.159, 40.81.28.194
Regional Identity Service - UK South	51.105.8.98

Next you will need to grant access to the data migration tool for Azure DevOps itself. You only need to grant an exception for the data migration tool instance in the region that you're importing into.

SERVICE	IP
Data migration tool - Central United States	52.173.74.9, 52.165.184.188, 20.45.1.234, 13.86.39.123
Data migration tool - West Europe	40.115.43.138, 13.95.15.128, 52.236.146.105, 40.67.219.89, 40.119.145.63, 52.142.236.228, 52.142.238.75
Data migration tool - Australia East	13.75.134.204, 40.82.219.41, 20.40.124.19
Data migration tool - Brazil South	104.41.24.164, 20.40.115.123
Data migration tool - India South	13.71.120.31, 40.81.76.137
Data migration tool - Canada Central	52.237.18.100, 52.237.24.61, 40.82.191.163
Data migration tool - East Asia (Hong Kong)	13.75.106.194, 40.81.27.181
Data migration tool - UK South	40.81.153.223

Next you will need to grant Azure DevOps Services access. Again, you only need to grant an exception for the Azure DevOps Services instance in the region that you're importing into.

SERVICE	IP
Azure DevOps Services - Central United States	13.89.236.72, 52.165.41.252, 52.173.25.16, 13.86.38.60, 20.45.1.175, 13.86.36.181, 52.158.209.56
Azure DevOps Services - West Europe	52.166.54.85, 13.95.233.212, 52.236.145.119, 52.142.235.223, 52.236.147.103, 23.97.221.25, 52.233.181.148, 52.149.110.153, 51.144.61.32, 52.236.147.236
Azure DevOps Services - Australia East	13.75.145.145, 40.82.217.103, 20.188.213.113, 104.210.88.194, 40.81.62.114

SERVICE	IP
Azure DevOps Services - Brazil South	20.40.114.3, 191.235.90.183, 191.232.38.181, 191.233.25.175
Azure DevOps Services - India South	104.211.227.29, 40.81.75.130, 52.172.54.122, 52.172.49.252
Azure DevOps Services - Canada Central	52.237.19.6, 40.82.190.38
Azure DevOps Services - East Asia (Hong Kong)	52.175.28.40, 40.81.25.218, 13.94.26.58
Azure DevOps Services - UK South	40.81.159.67

Next you will need to grant Azure Pipelines Releases service access. You only need to grant an exception for the Azure DevOps Services instance in the region that you're importing into.

Release Management IPs

SERVICE	IP
Releases service - United States	23.102.153.83, 23.101.127.247, 23.100.85.250, 13.86.39.233, 40.80.217.53, 52.232.229.122
Releases service - West Europe	13.95.223.69, 104.45.64.13
Releases service - Australia East	13.73.204.151, 20.40.176.135
Releases service - Brazil South	191.235.94.154, 20.40.116.69
Releases service - India South	52.172.15.233, 40.81.79.60
Releases service - Canada Central	52.237.28.171, 40.82.189.127
Releases service - East Asia (Hong Kong)	13.107.6.175, 40.81.29.43
Releases service - UK South	40.81.156.207

Next you will need to grant Azure Artifacts access. Again, you only need to grant an exception for the Azure DevOps Services instance in the region that you're importing into.

Azure Artifacts IPs

You will need to add exceptions for all three services that make up Azure Artifacts.

SERVICE	IP
Azure Artifacts - United States	52.173.148.93, 104.43.253.181, 23.99.179.148, 40.80.222.154, 40.119.0.130, 40.119.0.139, 13.86.125.169, 20.41.44.47, 40.90.219.165
Azure Artifacts - West Europe	104.46.45.12, 52.236.148.212
Azure Artifacts - Australia East	13.73.100.166, 20.40.176.15, 40.81.59.69

SERVICE	IP
Azure Artifacts - Brazil South	191.234.179.224, 20.40.115.214
Azure Artifacts - India South	52.172.11.191, 40.81.74.79
Azure Artifacts - Canada Central	52.237.24.224, 40.85.224.121, 13.71.189.199, 40.82.188.122
Azure Artifacts - East Asia (Hong Kong)	52.229.175.18, 65.52.162.53, 40.83.74.71, 40.81.27.130
Azure Artifacts - UK South	51.145.120.132

SERVICE	IP
Azure Artifacts Feed - United States	52.173.251.89, 20.45.1.3, 40.67.190.224, 20.41.58.125, 40.119.1.14, 20.45.1.249
Azure Artifacts Feed - West Europe	40.118.19.43, 52.236.146.118
Azure Artifacts Feed - Australia East	13.70.143.138, 20.40.176.80
Azure Artifacts Feed - Brazil South	191.235.93.87, 20.40.116.17
Azure Artifacts Feed - India South	52.172.8.41, 40.81.79.49
Azure Artifacts Feed - Canada Central	52.237.19.70, 40.82.188.254
Azure Artifacts Feed - East Asia (Hong Kong)	52.229.163.155, 40.81.28.59, 40.81.59.77
Azure Artifacts Feed - UK South	51.145.120.49

SERVICE	IP
Azure Artifacts Blob - United States	70.37.94.103, 40.78.129.25, 40.67.155.236, 52.230.216.163, 20.45.3.51
Azure Artifacts Blob - West Europe	23.97.221.25
Azure Artifacts Blob - Australia East	40.127.86.30, 20.188.213.113, 40.82.221.14
Azure Artifacts Blob - Brazil South	191.235.90.183
Azure Artifacts Blob - India South	52.172.54.122
Azure Artifacts Blob - Canada Central	52.237.16.145, 52.237.16.145, 52.233.38.115, 40.82.187.186
Azure Artifacts Blob - East Asia (Hong Kong)	13.94.26.58
Azure Artifacts Blob - UK South	51.143.174.59, 40.81.152.41

Analytics IPs (Azure DevOps Server 2019 or later only)

You only need to add an exception for the analytics IPs in your target import region if you included preview

features with your import.

SERVICE	IP
Analytics service - United States	20.41.43.22, 20.36.236.83, 20.41.40.50, 52.242.212.199, 13.86.33.148, 13.86.39.80
Analytics service - West Europe	52.236.146.143, 52.236.146.9
Analytics service - Australia East	20.40.179.159
Analytics service - Brazil South	20.40.113.248
Analytics service - India South	40.81.73.58
Analytics service - Canada Central	40.82.185.214
Analytics service - East Asia (Hong Kong)	40.81.25.239
Analytics service - UK South	40.81.159.247

Configuring IP Firewall Exceptions

Granting exceptions for the necessary IPs is handled at the Azure networking layer for your SQL Azure VM. To get started you will need to navigate to your SQL Azure VM on the [Azure portal](#). Then select 'Networking' from the settings. This will take you to the network interface page for your SQL Azure VM. The data migration tool requires the Azure DevOps Services IPs to be configured for inbound connections only on port 1433. Exceptions for the IPs can be made by selecting "Add inbound port rule" from the networking settings.



PROTOCOL	SOURCE	DESTINATION	ACTION	
TCP	Any	Any	<input checked="" type="checkbox"/> Allow	...
TCP	Any	Any	<input checked="" type="checkbox"/> Allow	...
Any	VirtualNetwork	VirtualNetwork	<input checked="" type="checkbox"/> Allow	...
Any	AzureLoadBalancer	Any	<input checked="" type="checkbox"/> Allow	...
Any	Any	Any	<input type="checkbox"/> Deny	...

Select advanced to configure an inbound port rule for a specific IP.

Add inbound security rule

X

 Advanced

Service i

Custom v

* Port range i

8080 

* Priority i

1510

* Name

Port_8080 

Description

Set the source to "IP Addresses", enter one of the IPs that need to be granted an exception, set the destination port range to 1433, and provide a name that best describes the exception you're configuring. Depending on other inbound port rules that have been configured, the default priority for the Azure DevOps Services exceptions might need to be changed so they don't get ignored. For example, if you have a deny on all inbound connections to 1433 rule with a higher priority than your Azure DevOps Services exceptions, the data migration tool might not be able to make a successful connection to your database.

* Source i

IP Addresses ▼

* Source IP address range i

168.62.105.45 ✓

* Source port range i

*

* Destination i

Any ▼

* Destination port range i

1433 ✓

* Protocol

Any TCP UDP

* Action

Allow Deny

* Priority i

1010 ✓

* Name

VSTS_Identity_Service ✓

Description

[Empty text area]

You will need to repeat adding inbound port rules until all necessary Azure DevOps Services IPs have been granted an exception. Missing one IP could result in your import failing to start.

Restoring your Database on the VM

After setting up and configuring an Azure VM, you will need to take your detached backup from your Azure DevOps Server instance to your Azure VM. Azure has several methods [documented](#) for how to accomplish this task. The collection database needs to be restored on SQL and doesn't require Azure DevOps Server to be installed on the VM.

Configuring your Collection for Import

Once your collection database has been restored onto your Azure VM, you will need to configure a SQL login to allow Azure DevOps Services to connect to the database to import the data. This login will only allow **read** access to a single database. Start by opening SQL Server Management Studio on the VM and open a new query window against the database that will be imported.

You will need to set the database's recovery to simple:

```
ALTER DATABASE [<Database name>] SET RECOVERY SIMPLE;
```

Next you will need to create a SQL login for the database and assign that login the 'TFSEEXECROLE':

```
USE [<database name>]
CREATE LOGIN <pick a username> WITH PASSWORD = '<pick a password>'
CREATE USER <username> FOR LOGIN <username> WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEEXECROLE', @membername='<username>'
```

Following our Fabrikam example the two SQL commands would look like the following:

```
ALTER DATABASE [Foo] SET RECOVERY SIMPLE;

USE [Foo]
CREATE LOGIN fabrikam WITH PASSWORD = 'fabrikamimport1!'
CREATE USER fabrikam FOR LOGIN fabrikam WITH DEFAULT_SCHEMA=[dbo]
EXEC sp_addrolemember @rolename='TFSEEXECROLE', @membername='fabrikam'
```

Configure the Import Specification File to Target the VM

The import specification file will need to be updated to include information on how to connect to the SQL instance. Open your import specification file and make the following updates:

Remove the DACPAC parameter from the source files object.

Before

```
"Source": {
  "Location": "<Provide the SASKey to the Azure storage container with the collection and
               import files.>",
  "Files": {
    "Dacpac": "Tfs_DefaultCollection.dacpac"
  }
},
```

After

```
"Source": {
  "Properties": {
    "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;
                       User ID=fabrikam;Password=fabrikam1!;Encrypt=True;TrustServerCertificate=True"
  }
},
```

Fill out the required parameters and add the following properties object within your source object in the specification file.

```
"Properties":
{
  "ConnectionString": "Data Source={SQL Azure VM IP};Initial Catalog={Database Name};Integrated
                      Security=False;User ID={SQL Login Username};Password={SQL Login
                      Password};Encrypt=True;TrustServerCertificate=True"
}
```

Following the Fabrikam example, the import specification would look like the following after applying the changes:

The screenshot shows a Visual Studio Code window with the title bar "import.json - Untitled (Workspace) - Visual Studio Code". The menu bar includes File, Edit, Selection, View, Go, Debug, Tasks, and Help. The left sidebar has icons for file, search, and refresh. The main editor area contains the following JSON code:

```
1  {
2      "Source": {
3          "Properties": {
4              "ConnectionString": "Data Source=8.8.8.8;Initial Catalog=Tfs_Foo;Integrated Security=False;User ID=fabrikam;Password=fabrikam!;Encrypt=True;TrustServerCertificate=True"
5          }
6      },
7      "Target": {
8          "Name": "fabrikam-import"
9      },
10     "Properties": {
11         "ImportType": "DryRun"
12     },
13     "ValidationData": {
14         "TfsMigratorVersion": "16.255.65000.0",
15         "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
16         "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
17         "DatabaseCollation": "SQL_Latin1_General_CI_AS",
18         "CommandExecutionCount": 0,
19         "CommandExecutionTime": 0.0,
20         "TfsVersion": "Dev15.M117",
21         "DatabaseTotalSize": 181,
22         "DatabaseBlobSize": 0,
23         "DatabaseTableSize": 181,
24         "DatabaseLargestTableSize": 8,
25         "ActiveUserCount": 8,
26         "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
27         "Region": "CUS",
28         "ValidationChecksumVersion": 1,
29         "ValidationChecksum": "66516G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
30     },
31     "Identities": [
32         "S-1-5-21-1374400868-3601225936-2087002269-500",
33         "S-1-5-21-2127521184-1604012920-1887927527-11008431",
34         "S-1-5-21-2127521184-1604012920-1887927527-15795496"
35     ]
36 }
```

The status bar at the bottom shows "Ln 1, Col 1" and "Spaces: 2" and "UTF-8 CRLF JSON".

Your import specification is now configured to use a SQL Azure VM for import! Proceed with the rest of preparation steps to import to Azure DevOps Services. Once the import has completed be sure to delete the SQL login or rotate the password. Microsoft does not hold onto the login information once the import has completed.

Uploading the DACPAC

NOTE

If you're using the SQL Azure VM method then you only need to provide the connection string. You will not have to upload any files and can skip this step.

Your DACPAC will need to be placed in an Azure storage container. This can be an existing container or one created specifically for your migration effort. It is important to ensure your container is created in the right region.

Azure DevOps Services is available in multiple [regions](#). When importing to these regions it's critical that you place your data in the correct region to ensure that the import can start successfully. Your data needs to be placed into the same region that you will be importing into. Placing it somewhere else will result in the import being unable to start. The below table covers the acceptable regions to create your storage account and upload your data.

DESIRED IMPORT REGION	STORAGE ACCOUNT REGION
Central United States	Central United States
Western Europe	Western Europe
Australia East	Australia East
Brazil South	Brazil South
India South	India South
Canada Central	Canada Central
East Asia (Hong Kong)	East Asia (Hong Kong)

While Azure DevOps Services is available in multiple regions in the United States, only the Central United States region is accepting new Azure DevOps Services. Customers will not be able to import their data into other United States Azure regions at this time.

[Creating a blob container](#) can be done from the Azure portal. Once the container has been created you will need to upload the following file:

- Collection DACPAC

After the import has been completed you can delete the blob container and accompanying storage account.

This can be accomplished using tools like [AzCopy](#) or any other Azure storage explorer tool like [Microsoft Azure Storage Explorer](#).

NOTE

If your DACPAC is larger than 10GB then it's recommended that you use AzCopy. AzCopy has multi-threaded upload support for faster uploads.

Generating SAS Key

A Shared Access Signature ([SAS](#)) Key provides delegated access to resources in a storage account. This allows you to give Microsoft the lowest level of privilege required to access your data for executing the import.

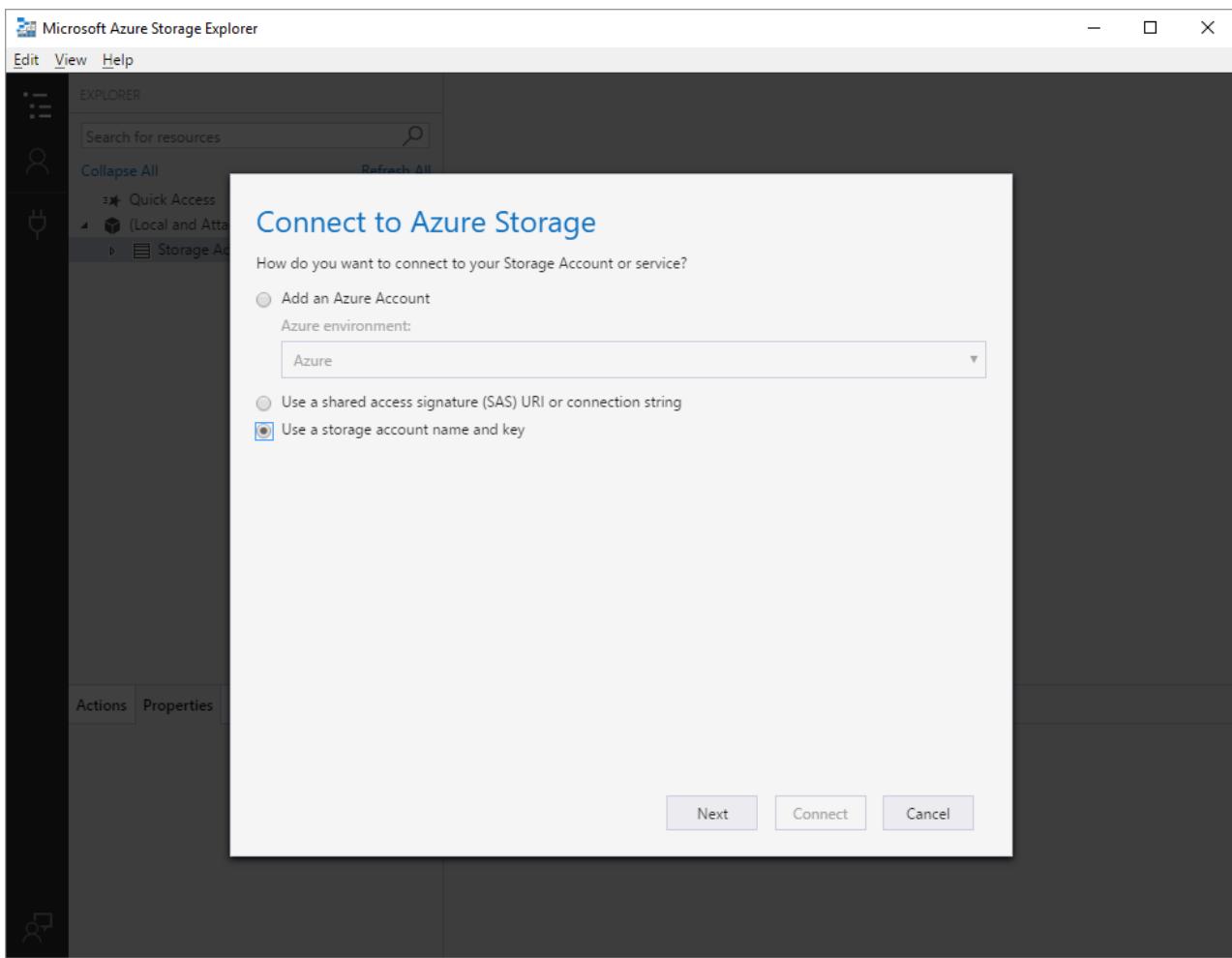
The recommended way to generate a SAS Key is the [Microsoft Azure Storage Explorer](#). Storage Explorer allows you to easily create container level SAS Keys. This is essential as the data migration tool does NOT support account level SAS Keys.

NOTE

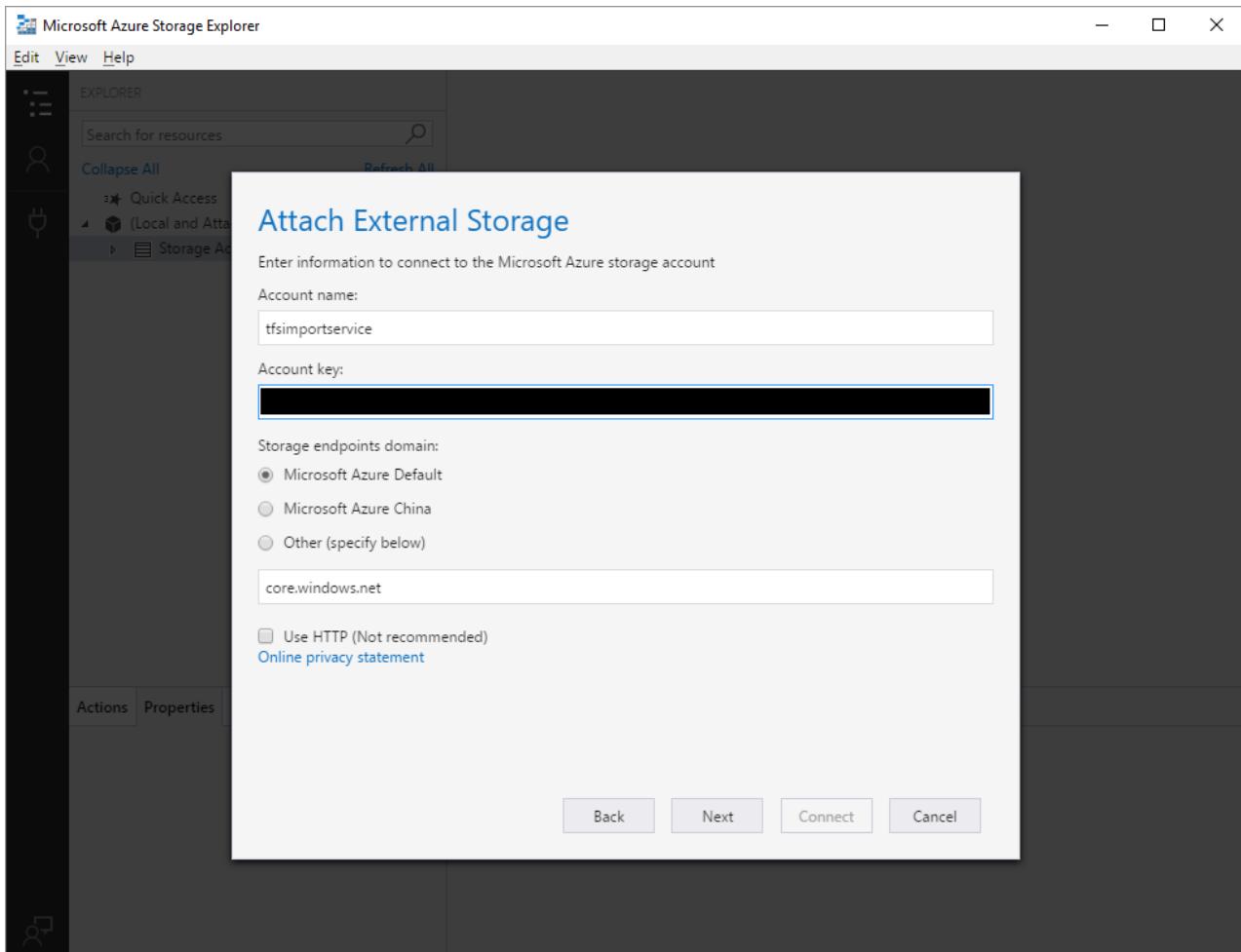
Do NOT generate a SAS Key from the Azure portal. Azure portal generated SAS Keys are account scoped and will not work with the data migration tool.

After installing Storage Explorer you can complete the following steps to generate a SAS Key:

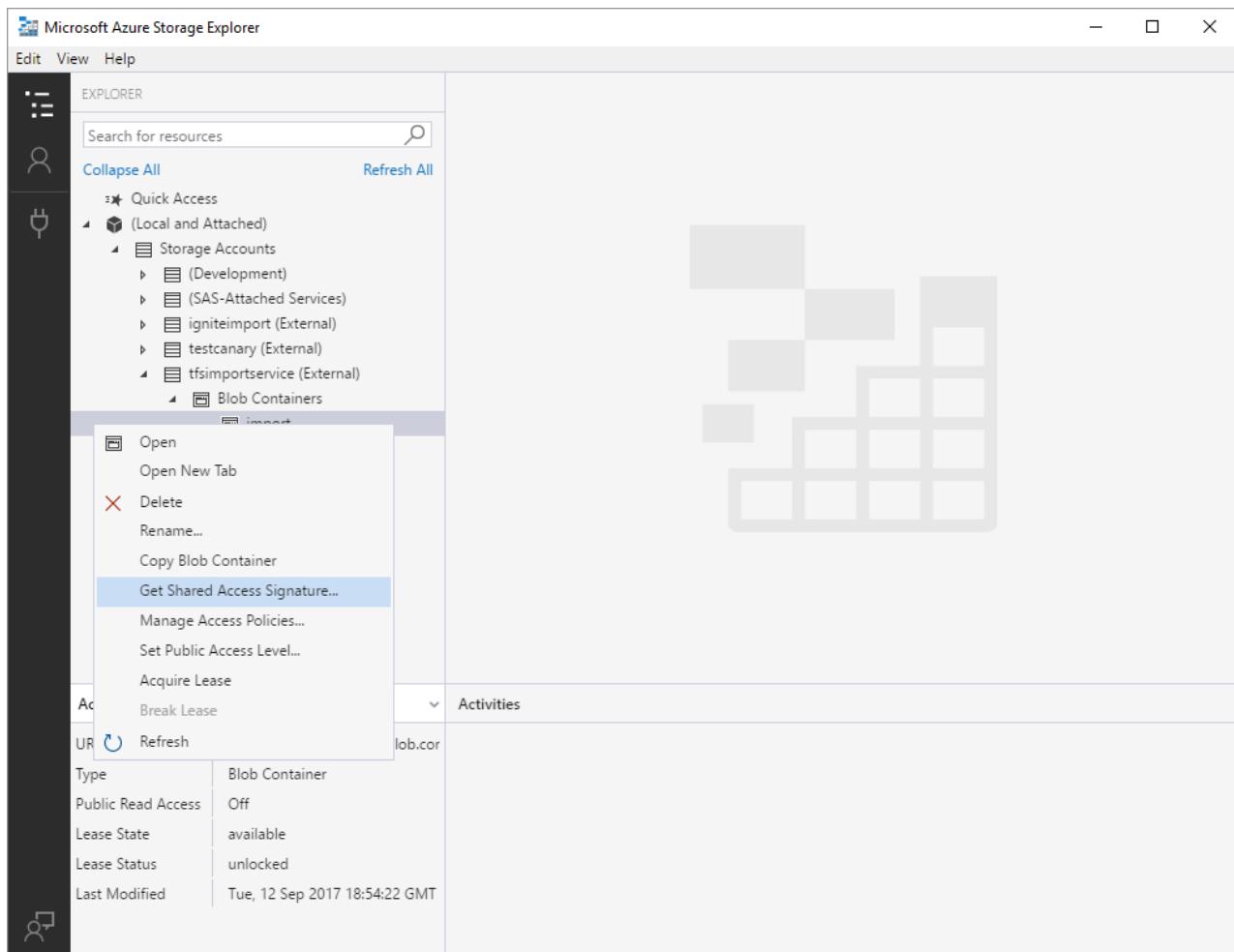
- Open the Microsoft Azure Storage Explorer after installation
- Add an account
- Choose "Use a storage account name and key"



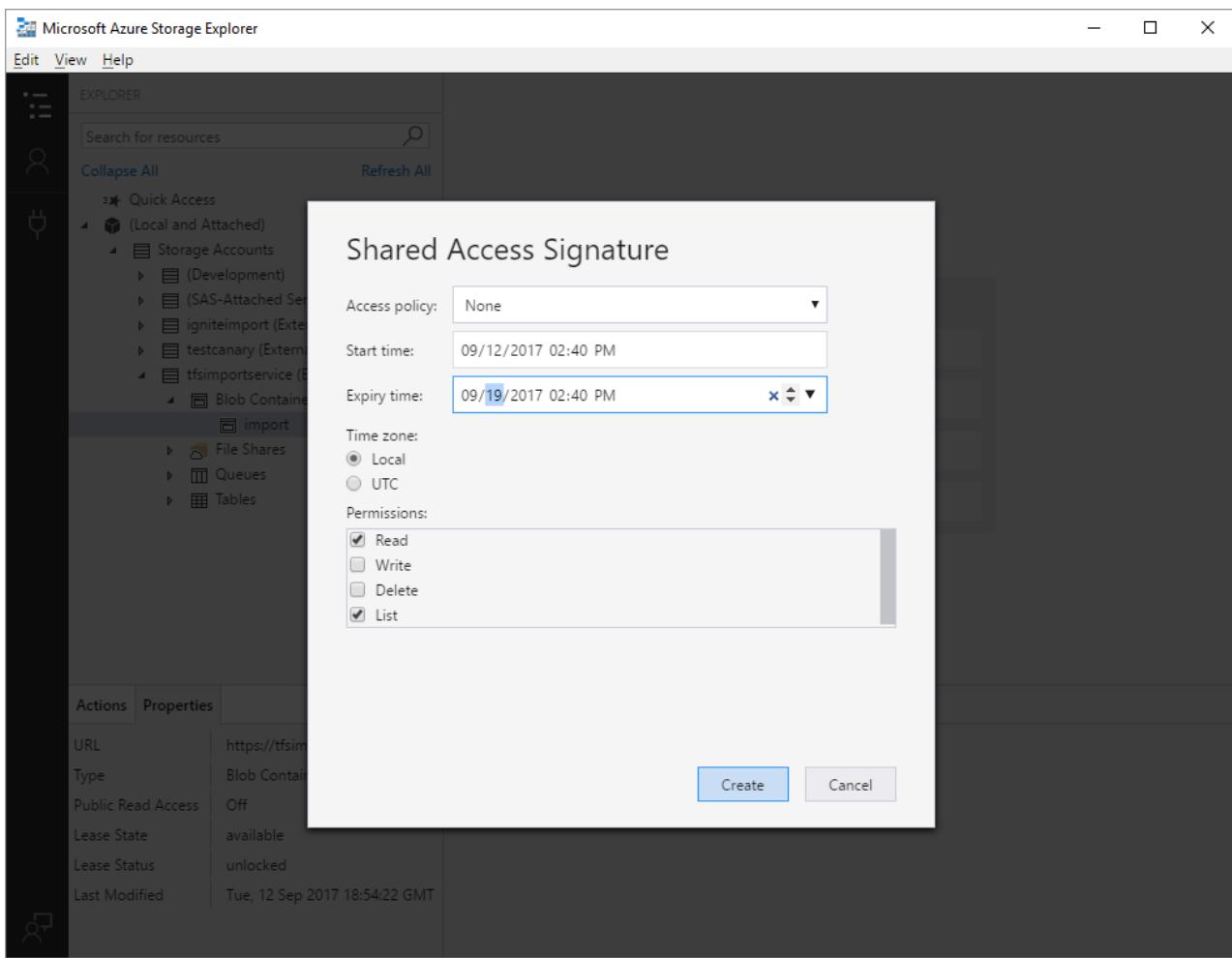
- Enter your storage account name, provide one of your two [primary access keys](#), and connect



- Expand out the blob containers, select the container with your import files, and choose to generate a "Shared Access Signature"



- Ensure that your SAS Key has read and list permissions. Write and delete permissions are NOT required
- Set the expiration for 7 days into the future



Copy and hold onto this SAS Key as you will need to place it in your import specification file during the next step.

Ensure that you treat this SAS Key as a secret. It provides access to your files in the storage container.

Completing the Import Specification

Earlier in the process you partially filled out the import specification file generally known as `import.json`. At this point you have enough information to fill out all of the remaining fields except for the import type. The import type will be covered in the import section below. Open your import specification file and fill out the following fields.

- **Location** - Place the SAS Key generated from the script in the last step here.
- **Dacpac** - Ensure the name in field is the same as the DACPAC file you uploaded to the storage account. Including the ".dacpac" extension.

Using the Fabrikam example, the final import specification file should look like the following:

```
1  "Source": {  
2      "Location": "https://fabrikam.blob.core.windows.net/fabrikam?st=2017-08-28T17%3A15%3A00Z&  
3          se=2017-08-29T17%3A15%3A00Z&sp=r&sv=2015-04-05&sr=c&  
4          sig=FSPF9sgifCFSmAWYz0icd009cxrmG9VP5pknFN16MgpY%3D",  
5      "Files": {  
6          "Dacpac": "Tfs_DefaultCollection.dacpac"  
7      },  
8  },  
9  "Target": {  
10     "Name": "fabrikam-import"  
11 },  
12  "Properties": {  
13      "ImportType": "<Provide the Type of Import: DryRun, ProductionRun>"  
14  },  
15  "ValidationData": {  
16      "TfsMigratorVersion": "16.255.65000.0",  
17      "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",  
18      "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",  
19      "DatabaseCollation": "SQL_Latin1_General_CI_AS",  
20      "CommandExecutionCount": 0,  
21      "CommandExecutionTime": 0.0,  
22      "TfsVersion": "Dev15.M117",  
23      "DatabaseTotalSize": 181,  
24      "DatabaseBlobSize": 0,  
25      "DatabaseTableSize": 181,  
26      "DatabaseLargestTableSize": 8,  
27      "ActiveUserCount": 8,  
28      "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",  
29      "Region": "CUS",  
30      "ValidationChecksumVersion": 1,  
31      "ValidationChecksum":  
32          "66S16G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="  
33  },  
34  "Identities": [  
35      "S-1-5-21-1374400868-3601225936-2087002269-500",  
36      "S-1-5-21-2127521184-1604012920-1887927527-11008431",  
37      "S-1-5-21-2127521184-1604012920-1887927527-15795496"  
38  ]  
39 }  
40  
41 1
```

0 0 1 1 json import.json Ln 1, Col 1 Spaces: 2 UTF-8 CRLF JSON

Determine the type of import

Imports can either be queued as a dry or production run. Dry runs are for testing and production runs are when your team intends to use the organization full time in Azure DevOps Services once the import completes.

Determining which type of import to be run is based off the value you provide for the import type parameter.

It's always recommended that you complete a dry run import first.

```
import.json - Untitled (Workspace) - Visual Studio Code
File Edit Selection View Go Debug Tasks Help
import.json x
1  [
2    "Source": {
3      "Location": "https://fabrikam.blob.core.windows.net/fabrikam?st=2017-08-28T17%3A15%3A00Z&se=2017-08-29T17%3A15%3A00Z&sp=r&sv=2015-04-05&sr=c&sig=FSPF9sgifCsmAWyZ0icd009cxrmG9VP5pknFN16MgpY%3D",
4      "Files": {
5        "Dacpac": "Tfs_DefaultCollection.dacpac"
6      }
7    },
8    "Target": {
9      "Name": "fabrikam-import"
10 },
11   "Properties": {
12     "ImportType": "DryRun"
13   },
14   "ValidationData": {
15     "TfsMigratorVersion": "16.255.65000.0",
16     "SourceCollectionId": "8b245d37-d41d-4188-a6f1-b5bb397860ba",
17     "DataImportCollectionId": "ca970402-9b06-4720-9407-ba32684e9499",
18     "DatabaseCollation": "SQL_Latin1_General_CI_AS",
19     "CommandExecutionCount": 0,
20     "CommandExecutionTime": 0.0,
21     "TfsVersion": "Dev15.M117",
22     "DatabaseTotalSize": 181,
23     "DatabaseBlobSize": 0,
24     "DatabaseTableSize": 181,
25     "DatabaseLargestTableSize": 8,
26     "ActiveUserCount": 8,
27     "TenantId": "72f988bf-86f1-41af-91ab-2d7cd011db47",
28     "Region": "CUS",
29     "ValidationChecksumVersion": 1,
30     "ValidationChecksum":
31       "66516G8u850KY6XKJm6MM5Ty3krNjhUFFCh4zyZMXqm7ZDLVpFpiIi0zDnJcoZmjHgDzvoCNS/9PwGm28hBgPg=="
32   },
33   "Identities": [
34     "S-1-5-21-1374400868-3601225936-2087002269-500",
35     "S-1-5-21-2127521184-1604012920-1887927527-11008431",
36     "S-1-5-21-2127521184-1604012920-1887927527-15795496"
37   ]
]
Ln 1, Col 1  Spaces: 2  UTF-8  CRLF  JSON  🌐
```

Dry Run Organizations

Dry run imports help teams to test the migration of their collections. It's not expected that these organizations will remain around forever, but rather to exist for a small time frame. In fact, before a production migration can be run, any completed dry run organizations will need to be deleted. All dry run organizations have a **limited existence and will be automatically deleted after a set period of time**. When the organization will be deleted is included in the success email received after the import completes. Be sure to take note of this date and plan accordingly.

Most dry run organizations will have 15 days before they're deleted. Dry run organizations can also have a 21 day expiration if more than 100 users are licensed basic or higher at **import time**. Once that time period passes the dry run organization will be deleted. Dry run imports can be repeated as many times as you need to feel comfortable before doing a production migration. A previous dry run attempt still needs to be deleted before attempting a new dry run migration. If your team is ready to perform a production migration before then you will need to manually delete the dry run organization.

Be sure to check out the [post import](#) article for additional details on post import activities. Should your import encounter any problems, review the [import troubleshooting](#) steps.

Run an import

The great news is that your team is now ready to begin the process of running an import. It's recommended that your team start with a dry run import and then finally a production run import. Dry run imports allow your team to see how the end results of an import will look, identify potential issues, and gain experience before heading into your production run.

NOTE

Repeating a production run import of a completed import for a collection, such as in the event of a rollback, requires reaching out to Azure DevOps Services [Customer Support](#) before queuing another import.

Considerations for Roll Back Planning

A common concern that teams have for the final production run is to think through what the rollback plan will be if anything goes wrong with import. This is also why we highly recommend doing a dry run to make sure you are able to test the import settings you provide to the data migration tool for Azure DevOps.

Rollback for the final production run is fairly simple. Before you queue the import, you will be detaching the team project collection from Team Foundation Server which will make it unavailable to your team members. If for any reason, you need to roll back the production run and have Team Foundation Server come back online for your team members, you can simply attach the team project collection on-premises again and inform your team that they will continue to work as normal while your team regroups to understand any potential failures.

Queueing an Import

IMPORTANT

Before proceeding, ensure that your collection was [detached](#) prior to generating a DACPAC or uploading the collection database to a SQL Azure VM. If you didn't complete this step the import will fail.

In the event your import fails, see the following [guidance](#).

Starting an import is done by using the data migration tool's import command. The import command takes an import specification file as input. It will parse through the file to ensure the values which have been provided are valid, and if successful, it will queue an import to Azure DevOps Services. The import command requires an internet connection, but does **NOT** require a connection to your Azure DevOps Server instance.

To get started, open a command prompt and CD to path where you have the data migration tool placed. Once there it's recommended that you take a second to review the help text provided with the tool. Run the following command to see the guidance and help for the import command:

```
Migrator import /help
```

The command to queue an import will have the following structure:

```
Migrator import /importFile:{location of import specification file}
```

Here is an example of a completed import command:

```
Migrator import /importFile:C:\DataMigrationToolFiles\import.json
```

Once the validation passes you will be asked to sign into to Azure AD. It's important that you sign in with an identity that is a member of the same Azure AD as the identity map log file was built against. The user that signs in will become the owner of the imported organization.

NOTE

Imports are limited to 5 against a single Azure AD tenant per 24 hour period. Only imports that are queued count against this cap.

After the import starts the user that queued the import will receive an email. Around 5-10 minutes after queueing the import your team will be able to navigate to the organization to check on the status. Once the import completes your team will be directed to sign in. The owner of the organization will also receive an email when the import finishes.

Process templates

10/2/2019 • 9 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server | TFS

NOTE

It's recommended that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019 the TFS Database Import Service has been rebranded to become data migration tool for Azure DevOps. This includes TfsMigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

The data migration tool could flag errors which need to be corrected before you start a migration. Below are the most common process related errors that you may encounter when preparing for a migration. After you have corrected each error, you will need to run the data migration tool's validate command again.

Process validation types

During the validation, the data migration tool will determine the target process model for each project. If the project was created with the Agile, Scrum, or CMMI process template, and was never customized, the project will use the Inheritance process model. In all other cases, the data migration tool considers the project as customized, and the project will use the Hosted XML process model. When the Hosted XML process is the targeted process model, the data migration tool validates if the customizations can be migrated. The data migration tool generates two files during the validation:

DataMigrationTool.log - Contains the set of process validation errors found in the collection. You must fix all these process errors to proceed with your migration.

TryMatchOobProcesses.log - Lists for each project the target process model - Inheritance or Hosted XML. For projects that are set to target the Hosted XML process model, it explains why they are considered to be customized. You don't have to fix these errors, but they give you guidance what to do in case you want to migrate into the Inheritance process model. Note that once a collection is imported, it is not possible to migrate a project to the Inheritance process model. That is on our roadmap for end of 2018.

Most customers have a mix of projects that have been customized (i.e. custom fields) and projects that are using an OOB process template. The data migration tool checks each project and validates it accordingly. It is very possible you will have some projects that will be mapped to an OOB process and some projects will use the Hosted XML for their process customization.

We recommend that for any project that has not been customized, that you review the TryMatchOobProcesses.log to determine if there are any errors. If so, make the adjustments accordingly so that the project can be mapped to an OOB process upon data import.

Update to a system process

If you started with an older version of Azure DevOps Server\TFS, odds are your projects are still using an older process template. If those projects have not been updated using the [Configure Features Wizard](#) then the data migration tool will find process errors. In some rare cases, if your process is so old, even the Configure Features

Wizard will not resolve the errors.

Here are some examples of error messages you will probably receive:

```
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element PortfolioBacklog is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element BugWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element FeedbackRequestWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402571: Required element FeedbackResponseWorkItems is missing from Process Configuration.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField Team.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField RemainingWork.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField Order.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField Effort.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField Activity.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField ApplicationStartInformation.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField ApplicationLaunchInstructions.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF402574: ProcessConfiguration doesn't specify required TypeField ApplicationType.  
Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:: TF400572: The Project Process Settings must be configured for this feature to be used.
```

If you have never customized your project (added fields, work item types, etc.), then fixing these is actually pretty simple.

If you customized your process, then this approach will not work. You will need to manually change the process templates so that your customizations do not get overwritten.

First, make sure you know what process your project started as. Is it Scrum, Agile or CMMI? In this example, let us assume Agile. Now go to <https://github.com/Microsoft/process-customization-scripts> and download the repo. In this instance, we are going to focus on contents in the "Import" folder.

Use the "ConformProject.ps1" script to conform a project of your choosing to the Agile system process. This will update the entire project to be Agile.

```
. \ConformProjects.ps1 "<collection url>" "<project name>" "c:\process-customization-scripts\import\agile"
```

Make sure you do this for each and every project.

Resolve process errors

Are your process templates customized? Are you using an older outdated process template? If so, you will most likely have process validation errors. The data migration tool does an exhaustive check against your process templates. It checks to make sure that it is valid for Azure DevOps Services. Odds are you will need to make some adjustments and apply them to your collection.

NOTE

If you are using an OOB Agile, Scrum, or CMMI process you probably won't see any errors in the DataMigrationTool.log. Instead, check the TryMatchOobProcesses.log for errors. If you are error free, then your project will map to an OOB process.

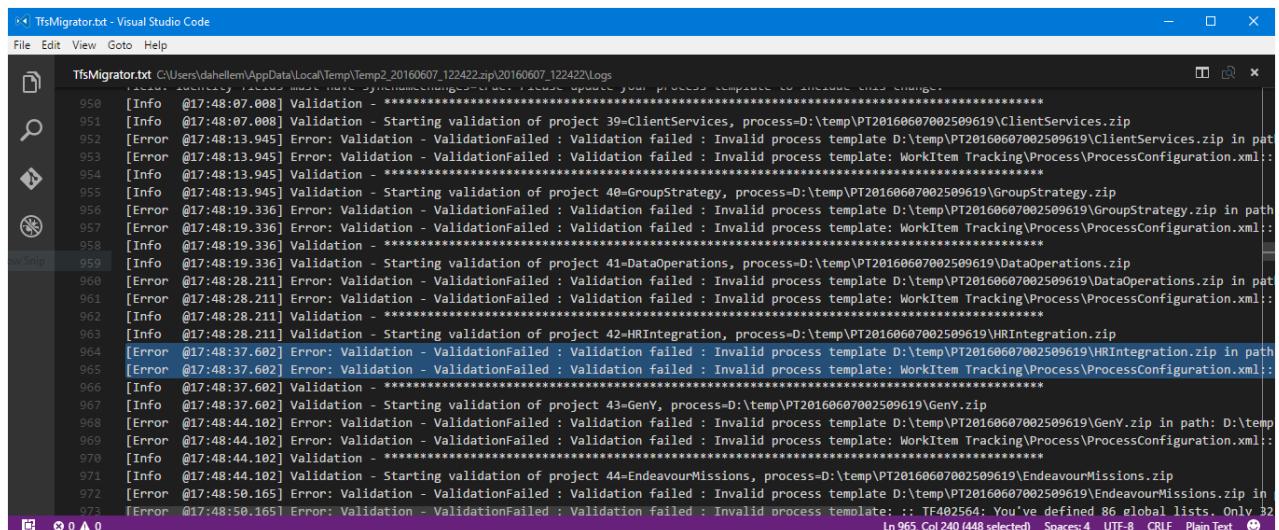
There are variety of customizations that will not work in Azure DevOps Services. Make sure you review the [list of customizations](#) that are supported.

If you have projects that are using an older process template, the data migration tool will find several errors. This is because your process templates have not been updated to match the most recent process templates. To start, try running the [Configure Features Wizard](#) for each project. This will attempt to update your process templates with the most recent features. Doing so should drastically reduce the error count.

Finally, make sure you have [witadmin](#) on the machine that you intend to use to fix the process errors. This can be your local desktop. Witadmin is used in the automated scripts and is required whenever making changes to the process templates.

Step 1 - Review errors

DataMigrationTool.log file will be generated and contains the list of errors that the validation process found. To view the logs, open DataMigrationTool.log file. Search for the string "Validation - Starting validation of project 1". Each project is validated so you will need to scan through all the projects. Examine any lines that have a prefix of "[Error ...]."



```
TfsMigrator.txt - Visual Studio Code
File Edit View Goto Help
TfsMigrator.txt C:\Users\dahellen\AppData\Local\Temp\Temp2_20160607_122422\Logs
[Info @17:48:07.008] Validation - *****
[Info @17:48:07.008] Validation - Starting validation of project 39=ClientServices, process=D:\temp\PT20160607002509619\ClientServices.zip
[Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\ClientServices.zip in path
[Error @17:48:13.945] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
[Info @17:48:13.945] Validation - *****
[Info @17:48:13.945] Validation - Starting validation of project 40=GroupStrategy, process=D:\temp\PT20160607002509619\GroupStrategy.zip
[Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GroupStrategy.zip in path
[Error @17:48:19.336] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
[Info @17:48:19.336] Validation - *****
[Info @17:48:19.336] Validation - Starting validation of project 41=DataOperations, process=D:\temp\PT20160607002509619\DataOperations.zip
[Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\DataOperations.zip in path
[Error @17:48:28.211] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
[Info @17:48:28.211] Validation - *****
[Info @17:48:28.211] Validation - Starting validation of project 42=HRIntegration, process=D:\temp\PT20160607002509619\HRIntegration.zip
[Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\HRIntegration.zip in path
[Error @17:48:37.602] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
[Info @17:48:37.602] Validation - *****
[Info @17:48:37.602] Validation - Starting validation of project 43=GenY, process=D:\temp\PT20160607002509619\GenY.zip
[Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GenY.zip in path: D:\temp
[Error @17:48:44.102] Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
[Info @17:48:44.102] Validation - *****
[Info @17:48:44.102] Validation - Starting validation of project 44=EndeavourMissions, process=D:\temp\PT20160607002509619\EndeavourMissions.zip
[Error @17:48:50.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\EndeavourMissions.zip in path
[Error @17:48:50.165] Error: Validation - ValidationFailed : Validation failed : Invalid process template: :: TFA9256A: You've defined 86 global lists. Only 32
Ln 965, Col 240 (448 selected) Spaces:4 UTF-8 CRLF Plain Text
```

We have documented the majority of the [validation errors](#). For each validation error we have provided the error number, description, and the method to resolve.

Step 2 - Fix errors

Now you know what projects have errors, the details of those errors, and how to fix them. Fixing the errors requires that you to change the xml and apply the changes back into the project.

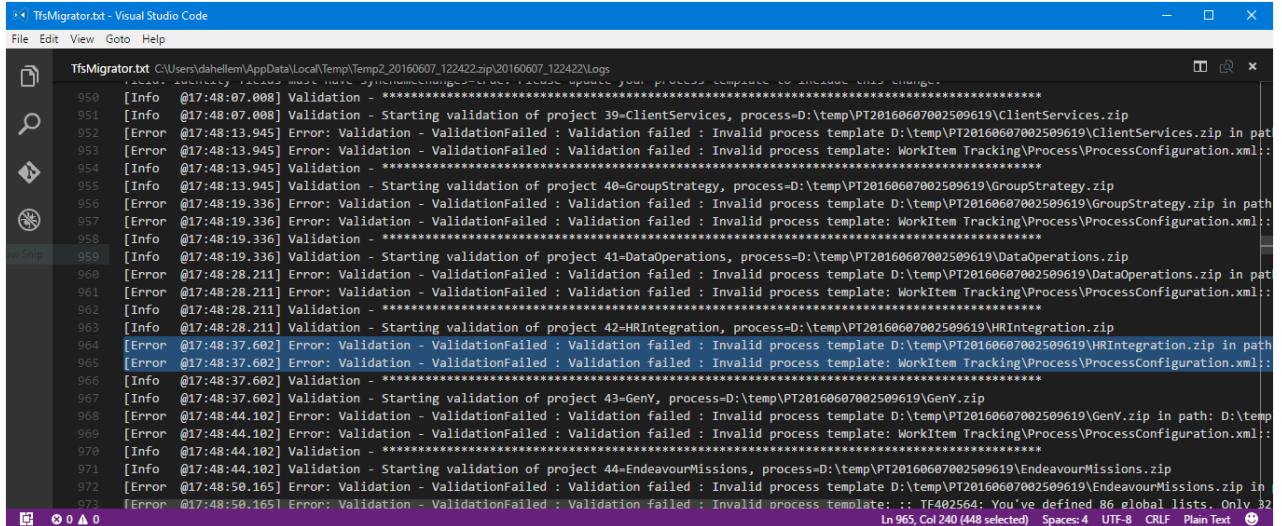
We do not suggest using the TFS Power Tools. It is highly recommended that you modify the XML manually.

To get the process template from the project add the /SaveProcesses parameter when running the data migration tool command.

```
Migrator validate /collection:{collection URL} /SaveProcesses
```

This command will extract the xml from the project and place it into the same folder as the logs. Extract the zip files to your local machine so that you can edit the files.

Now you need to fix the xml. Use the logs from the `DataMigrationTool.log` file to determine the errors for each project.



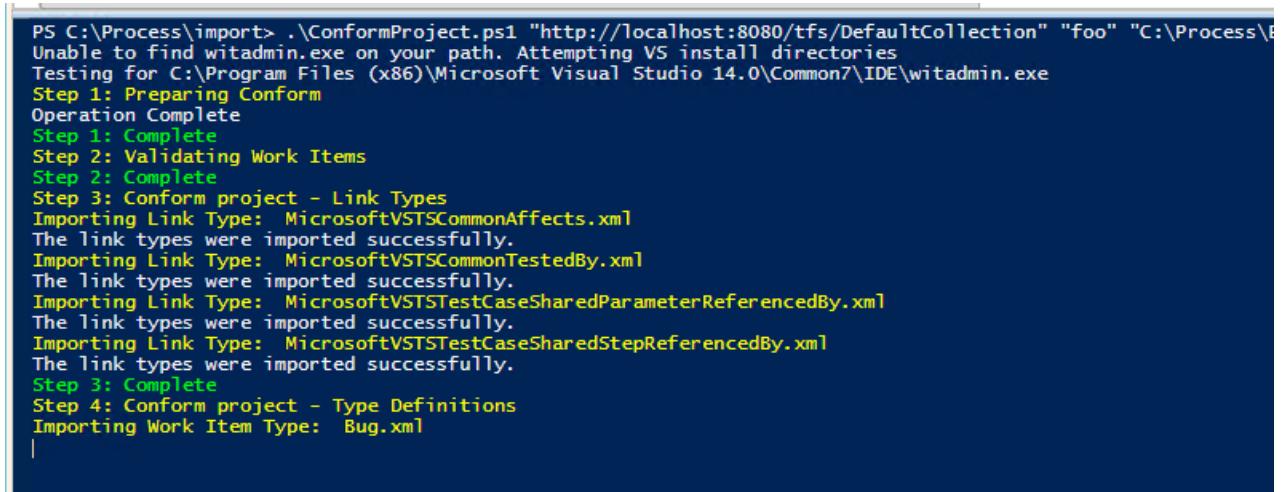
```
TfsMigrator.txt - Visual Studio Code
File Edit View Goto Help
TfsMigrator.txt C:\Users\dahellew\AppData\Local\Temp\Temp2_20160607_122422.zip\20160607_122422\Logs
950 [Info] @17:48:07.008 Validation - ****
951 [Info] @17:48:07.008 Validation - Starting validation of project 39=ClientServices, process=D:\temp\PT20160607002509619\ClientServices.zip
952 [Error] @17:48:13.945 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\ClientServices.zip in path
953 [Error] @17:48:13.945 Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
954 [Info] @17:48:13.945 Validation - ****
955 [Info] @17:48:13.945 Validation - Starting validation of project 40=GroupStrategy, process=D:\temp\PT20160607002509619\GroupStrategy.zip
956 [Error] @17:48:19.336 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GroupStrategy.zip in path
957 [Error] @17:48:19.336 Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
958 [Info] @17:48:19.336 Validation - ****
959 [Info] @17:48:19.336 Validation - Starting validation of project 41=DataOperations, process=D:\temp\PT20160607002509619\DataOperations.zip
960 [Error] @17:48:19.336 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\DataOperations.zip in path
961 [Error] @17:48:28.211 Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
962 [Info] @17:48:28.211 Validation - ****
963 [Info] @17:48:28.211 Validation - Starting validation of project 42=HRIntegration, process=D:\temp\PT20160607002509619\HRIntegration.zip
964 [Error] @17:48:37.602 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\HRIntegration.zip in path
965 [Error] @17:48:37.602 Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
966 [Info] @17:48:37.602 Validation - ****
967 [Info] @17:48:37.602 Validation - Starting validation of project 43=GenY, process=D:\temp\PT20160607002509619\GenY.zip
968 [Error] @17:48:44.102 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\GenY.zip in path: D:\temp
969 [Error] @17:48:44.102 Error: Validation - ValidationFailed : Validation failed : Invalid process template: WorkItem Tracking\Process\ProcessConfiguration.xml:::
970 [Info] @17:48:44.102 Validation - ****
971 [Info] @17:48:44.102 Validation - Starting validation of project 44=EndeavourMissions, process=D:\temp\PT20160607002509619\EndeavourMissions.zip
972 [Error] @17:48:50.165 Error: Validation - ValidationFailed : Validation failed : Invalid process template D:\temp\PT20160607002509619\EndeavourMissions.zip in path
973 [Error] @17:48:50.165 Error: Validation - ValidationFailed : Validation failed : Invalid process template: :: TF402564: You've defined 86 global lists. Only 32
Ln 965, Col 240 (448 selected) Spaces:4 UTF-8 CRLF Plain Text ☺
```

Some errors will require you to do use a `witadmin changefield` command. Changing a field name is the most common example. To save yourself some time, we recommend you run the `witadmin changefield ...` command and then re-run the data migration tool tool. Doing this will re-export the xml with the corrected names. Otherwise you will need manually fix the fields in the xml as well.

Once you make a fix then you need to conform. Conform is defined as taking the XML you just changed and applying it back into Azure DevOps Server. To do this, depending on the changes you made, you will need to run one or more `witadmin` commands. To make this easier for you, we created a PowerShell script to automate the process. The script contains all of the `witadmin` commands needed to conform the entire process.

You can get the scripts at <https://github.com/Microsoft/process-customization-scripts>. Use the import/ConformProject.ps1 script.

```
.\conformproject.ps1 "<collection url>" "<project name>" "<process template folder>"
```



```
PS C:\Process\import> .\ConformProject.ps1 "http://localhost:8080/tfs/DefaultCollection" "foo" "C:\Process\Import"
Unable to find witadmin.exe on your path. Attempting VS install directories
Testing for C:\Program Files (x86)\Microsoft Visual Studio 14.0\Common7\IDE\witadmin.exe
Step 1: Preparing Conform
Operation Complete
Step 1: Complete
Step 2: Validating Work Items
Step 2: Complete
Step 3: Conform project - Link Types
Importing Link Type: MicrosoftVSTSCommonAffects.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSCommonTestedBy.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSTestCaseSharedParameterReferencedBy.xml
The link types were imported successfully.
Importing Link Type: MicrosoftVSTSTestCaseSharedStepReferencedBy.xml
The link types were imported successfully.
Step 3: Complete
Step 4: Conform project - Type Definitions
Importing Work Item Type: Bug.xml
|
```

When the script has completed you need to re-run the data migration tool to validate the collection. Follow steps 1 - 3 until the data migration tool generates no more validation errors.

If you are new to xml and `witadmin`, we suggest you make one fix at a time and then conform. Continue this loop until all errors are resolved.

Common validation errors

VS402841: Field X in work item type Bug has syncnamechanges=false but has rules making it an identity field. Identity fields must have

syncnamechanges=true. Please update your process template to include this change.

In Azure DevOps Services we added a rule so that every identity field must have the syncnamechanges=true attribute. In AzureDevOpsServer that rule does not apply. Therefore, the data migration tool will identify this as an issue. Don't worry, making this change on Azure DevOps Server on-prem will not cause any harm.

To fix this you will need to run the witadmin changefield command. Syntax for the command will look something like this:

```
witadmin changefield /collection:http://AdventureWorksServer:8080/tfs/DefaultCollection /n:fieldname  
/syncnamechanges:true
```

For more information on the changefield command see <https://msdn.microsoft.com/library/dd236909.aspx>

TF402556: For field System.IterationId to be well defined, you must name it Iteration ID and set its type to Integer.

This error is typical for old process templates that have not been updated in some time. Try running the [configure features wizard](#) on each project. Alternatively you can run the follow witadmin command:

```
witadmin changefield /collection:http://AdventureWorksServer:8080/tfs/DefaultCollection /n:fieldname  
/name:newname
```

TF402571: Required element BugWorkItems is missing from Process Configuration.

This error typically occurs when a process has not been updated in a while. Try running the [configure features wizard](#) on each project to resolve.

TF402564: You've defined XX global lists. Only 64 are allowed.

By default, Azure DevOps Services will support 64 global lists. You will typically run across this error if you have a large amount of build pipelines. The global list named Builds - **TeamProjectName** gets created for each new build pipeline. You will need remove the outdated global lists.

Related articles

- [witadmin](#)
- [Differences between Azure DevOps Services and Azure DevOps Server process template customizations](#)
- [Configure features after Azure DevOps Server upgrade](#)
- [Resolve validation errors](#)
- [Define global lists in Azure DevOps Server](#)
- [Process customization PowerShell scripts](#)

Post import

10/2/2019 • 5 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server](#) | [TFS](#)

NOTE

It's recommended that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019 the TFS Database Import Service has been rebranded to become data migration tool for Azure DevOps. This includes TfsMigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

An organization is ready for use once an import has completed successfully. However, there are common tasks that you should perform before opening the organization up to all of your users. Below is a list of the most common after import tasks that should be completed. Tasks are listed in recommended order of completion.

Immediately after import

Immediately after the organization becomes available you will want to take a small team and perform spot checks on the organization. It's recommended that this team consists of the project collection administrators. This shouldn't be an in-depth check, but rather making sure that major pieces from your collection were brought over. Did your source code get imported? Are you seeing your build history? Are all of our area paths still present? It's best to confirm these artifacts are present before opening the organization to the entirety of your user base.

After spot checking the organization you will want to consider if you want to rename it. [Renaming an organization](#) is a simple operation, but it has [large impacts](#) on users currently using the organization. Some examples being Team Explore connections breaking or bookmarks no longer working. Getting a rename out of the way while it's just a small group of users using the organization allows the rest of the users to come in and configure their connections once.

Set up billing

To pay for users or services in Azure DevOps Services, like hosted build and deployment agents, you'll need to set up billing for your organization. You can do this when you make your first purchase by selecting an Azure subscription that you'd like to use for billing. This links the subscription to your organization, so that all future purchases for your organization will use the same Azure subscription for billing.

Manage users and access

Your organization includes 5 free users with [Basic](#) access. Basic includes features like Git and Team Foundation version control, tools for agile planning and Java teams, and more. Also, you can add [Visual Studio subscribers](#) for free - they get basic features plus additional features, based on their subscription level. Add [Stakeholder](#) for free, too-they can access your work items and view your backlogs.

This means that you shouldn't have to take any other import steps if your identity mapping file has just 5 users with Basic access, Visual Studio subscriptions, and Stakeholder access. If you have more than 5 users with Basic access, you'll need to [pay for these users in your organization](#). Just make sure to do this before the end of the

calendar month when you import. Otherwise, these users' feature access will change from Basic to Stakeholder on the 1st day of the next calendar month. To find out how many additional users you'll need to pay for, visit your organization (https://dev.azure.com/{yourorganization}/_user) so you can find the number of paid users that you've assigned:

User Summary	
Free	5
Paid	7
Visual Studio Enterprise	1
Total	13

Dry run imports do not have their licenses reset on the 1st of the month. Unlike production imports, their grace period extends for the life of the organization. So you don't need to worry about purchasing licenses while testing out an import.

Builds

Next, you will want to configure your build agents. As part of the migration, all of your build pipelines have been brought over, but agents and pools need to be reconfigured against the new organization. Azure DevOps Services offers the ability to use a Microsoft-hosted pool of build agents that you can use, or you can connect your self-hosted build agent(s). It's important to note that only one self-hosted build agent is included for free. After that there is a [fee](#) for having additional self-hosted build agents. To pay for Microsoft-hosted and self-hosted build agents you will need to link a subscription to your organization. See the following resources for details on performing this task:

- [Build Agents](#)

If you plan on using your existing on-premises private build agents, there is one more recommended step that needs to be taken after registering them to your new organization. Clearing their cache will ensure that you don't encounter any build issues related to older TFVC or Git pointers to your on-premises collection. See [refreshing caches on client computers](#) for details on how to accomplish this task.

Release management

If you used Release Management in Azure DevOps Server then your release pipelines and history data will be included with your import. However, like builds, [agents](#) and pools need to be reconfigured against the new organization.

Azure Artifacts

If you used Azure Artifacts in your collection, then you will need to install the Azure Artifacts [extension](#) in your organization post import to view your Azure Artifacts data.

Azure Boards

If you have an existing GitHub Enterprise Server connection associated with your Azure DevOps Server, it will not work as expected. Work item mentions within GitHub may be delayed or never show up in Azure DevOps Services. This problem occurs because the callback url associated with GitHub is no longer valid.

To resolve the problem, consider the following:

- **Remove and re-create the connection:** Remove and re-create the connection to the GitHub Enterprise Server repository. Follow the sequence of steps provided in [Connect from Azure Boards](#) documentation.

- **Fix the webhook url:** Go to GitHub's repository settings page and edit the webhook url to point out to the migrated Azure DevOps Services organization url:

```
https://dev.azure.com/{OrganizationName}/_apis/work/events?api-version=5.2-preview
```

Notify your teams

After getting your builds running and license subscription configured, it's recommended that the organization be opened up to all users for validation. This is when individual users can ensure that all of the content is in place, they have the right access level, and that they can pull code. Be sure to point users to our [documentation](#) on connecting to Azure DevOps Services from all of our supported IDEs and Team Explorer.

Users of TFVC with local workspaces will need to remap their workspaces against the new organization and Git users will have to reconfigure their remotes to be able to pull code.

If anything is reported as missing from the migrated organization, please reach out to AzureDevOpsImport@microsoft.com. For other functional issues, please reach out to [customer support](#).

Troubleshooting

10/2/2019 • 19 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server | TFS

NOTE

It's recommended that you use the [Migration Guide](#) to progress through your import. The guide links to the technical documentation as needed.

With the release of Azure DevOps Server 2019 the TFS Database Import Service has been rebranded to become data migration tool for Azure DevOps. This includes TfsMigrator becoming the data migration tool or migrator for short. This service still works exactly the same as the old Import Service. If you're on an older version of on-premises with TFS as the branding you can still use this feature to migrate to Azure DevOps as long as you upgrade to one of the supported versions.

The data migration tool could flag errors which need to be corrected prior to performing a migration. Below are the most common errors that are encountered when prepping for a migration. After correcting each error you will need to run the data migration tool validate command again to ensure the error(s) is/are actually gone.

Resolve size warnings

If your collection is particularly large then you might receive one of the below messages after running the data migration tool. If you receive any of the below warnings or errors, it's always recommended that you try to [reduce your database's size](#).

The database is currently {Database Size}GBs. This is above the recommended size of {DACPAC Size Limit}GBs to use the DACPAC import method. Please see the following page to learn how to import using a SQL Azure VM:
<https://aka.ms/AzureDevOpsImportLargeCollection>

This is a warning that means you will need to use the SQL Azure VM method to complete your import. Once a database reaches a certain size it becomes faster to setup a SQL Azure VM to complete the import to Azure DevOps Services. Follow the instructions linked from the warning message to setup the VM and complete your import. This warning does **NOT** mean that your collection is too big to be imported.

The largest table size is currently {Table size}GBs. This is above the recommended size of {Size limit}GBs to use the DACPAC import method. Please see the following page to learn how to import using a SQL Azure VM:
<https://aka.ms/AzureDevOpsImportLargeCollection>

Similar to the previous warning, this warning means you will have to use the SQL Azure VM method to complete the import. Follow the instructions linked from the warning message to setup the VM and complete your import. This warning does **NOT** mean that your collection is too big to be imported.

The database metadata size is currently {Metadata Size}GBs. This is above the recommended size of {Warning Size}GBs. It's recommended that you consider cleaning up older data as described in [Cleaning up old data] (/azure/devops/server/upgrade/clean-up-data).

This warning means that your database is approaching the limit for total metadata size. Metadata size refers to the size of your database without including files, code, and other binary data. The warning does **NOT** mean that your collection is too big for import, rather its metadata size is larger than the vast majority of other databases.

It's strongly recommended that you [reduce the size](#) of your database before import. Reducing the size provides the additional benefit of speeding up your import.

The database metadata size is currently {Metadata Size}GBs. This is above the maximum supported size of {Metadata Limit}GBs.

Unlike the previous warnings, this is an error that **WILL** block you from moving forward with your migration to Azure DevOps Services. The volume of metadata in your collection is too large and needs to be [reduced](#) below the mentioned limit to proceed with the import.

Resolve collation warnings

Collation in this case refers to the collection database's collation. Collations control the way string values are sorted and compared. Collections that aren't using either SQL_Latin1_General_CI_AS or Latin1_General_CI_AS will generally receive one of the two below **warning** messages.

The collection database's collation '{collation}' is not natively supported in Azure DevOps Services. Importing your collection will result in your collation being converted to one of the supported Azure DevOps Services collations. See more details at <https://aka.ms/AzureDevOpsImportCollations>

Receiving this warning **does NOT** mean that you can't import your collection to Azure DevOps Services. Rather, it means that you will need to think through some additional considerations before performing an import. When a non-supported collation is imported into Azure DevOps Services it is effectively transformed to the supported Azure DevOps Services collation. While this generally works without issue, unexpected results could be observed post import or the import could fail if a unique collation translation issue is encountered. For instance, customers will notice different ordering for strings containing non-English characters. Non-English characters like 'é' may become equivalent to the English 'e' after the import has completed. It's important that you complete and verify a dry run import when importing a collection with a non-supported collation.

This warning requires an acknowledgement from the user running the data migration tool command. Accepting the warning will allow the data migration tool to continue assisting you with preparing for your import.

The collections database's collation '{collation}' is not natively supported in Azure DevOps Services. It could not be validated that the collation can be converted during import to a supported Azure DevOps Services collation, as there was no internet connection. Please run the command again from a machine with an internet connection. See more details at <https://aka.ms/AzureDevOpsImportCollations>

If the data migration tool is unable to make a connection to the internet then it will be unable to validate that your collation can be converted to one of the supported version at import time. It's only a warning, so you will be able to make forward progress on your migration process. However, when you run the prepare command, an internet connection is required and your collation will be validated at that time.

Generally a non-supported collation can be converted to one of the supported collations at import time. However, in extreme cases there are some collations which can't be converted. If your collection uses one of those collations then you will receive the below **error** message.

The collection database's collation '{collation}' is not supported for import to Azure DevOps Services. It will need to be changed to a supported collation before it can be imported. See more details at <https://aka.ms/AzureDevOpsImportCollations>

In order to continue your collection's collation will need to be [changed](#) to one of the supported collations on Azure DevOps Services.

Resolve identity errors

Identity errors aren't common when validating a collection, but when they do come up it's important to fix them prior to migration to avoid any undesired results. Generally, identity problems stem from valid operations on previous versions of TFS that are no longer valid on your current Azure DevOps Server version. For example, some users being members of a built-in valid users group was once allowed, but isn't in more recent versions. The most common identity errors and guidance on fixing them can be found below.

ISVError:100014

This error indicates that a permission is missing from a system group. System groups are well known groups in Azure DevOps Server and Azure DevOps Services. For example, every collection that you create has "Project Collection Valid Users" and "Project Collection Administrators" groups. They're created by default and it's not possible to edit the permissions for these groups. What this error indicates is that somehow one or more of these groups is missing a permission that it's expected to have. In order to fix this, you will need to use TFSSecurity.exe to apply the expected permissions onto the flagged system groups. To get started you will need to identify which [TFSSecurity](#) command(s) will need to be run.

Project Collection Valid Users error message

Carefully examine the error message(s) the data migration tool highlighted. If the group that was flagged ends with "**0-0-0-0-3**", such as in the example below, then you will need to fix a missing permission for the "Project Collection Valid Users" group. Run the below command against TFSSecurity.exe after replacing the scope with the one from the error message and adding in your collection URL.

```
TFSSecurity.exe /a+ Identity "{scope}\\" Read sid:{Group SID} ALLOW /collection:{collectionUrl}
```

In the below example you will need to take the scope and group SID from the error message, and add it the templated command above.

```
ISVError:100014 Missing permission for group:Microsoft.TeamFoundation.Identity;S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-3 for scope:397c326b-b97c-4510-8271-75aac13de7a9. Expected:1 and Actual:0
```

The final command will look like:

```
TFSSecurity.exe /a+ Identity "397c326b-b97c-4510-8271-75aac13de7a9\\" Read sid:S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-3 ALLOW /collection:https://localhost:8080/defaultcollection
```

Project Collection Administrators error message

Carefully examine the error message(s) the data migration tool highlighted. If the group that was flagged ends with "**0-0-0-0-1**", such as in the example below, then you will need to fix a missing permission for the "Project Collection Administrators" group. Run the below commands against TFSSecurity.exe after replacing the scope with the one from the error message and adding in your collection.

```
TFSSecurity.exe /a+ Identity "{scope}\\" Read sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\" Write sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\" Delete sid:{Group SID} ALLOW /collection:{collectionUrl}  
TFSSecurity.exe /a+ Identity "{scope}\\" ManageMembership sid:{Group SID} ALLOW /collection:{collectionUrl}
```

In the below example you will need to take the scope and group SID from the error message, and add it the templated command above.

```
ISVError:100014 Missing permission for group:Microsoft.TeamFoundation.Identity;S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-0-1 for scope:0c7c2216-fa4b-4107-a203-82b324a147ef. Expected:15 and Actual:0
```

The final command will look like:

```
TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\" Read sid:S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection

TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\" Write sid:S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection

TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\" Delete sid:S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection

TFSSecurity.exe /a+ Identity "0c7c2216-fa4b-4107-a203-82b324a147ef\\\" ManageMembership sid:S-1-9-XXXXXXXXXX-XXXXXXX-XXXXXXX-XXXXXXX-0-0-0-0-1 ALLOW /collection:https://localhost:8080/defaultcollection
```

If you have multiple errors that need to be corrected, it's recommended that you put all of the commands into a batch file to execute them against TFSSecurity in an automated fashion. Once the commands have been executed you will need to run the data migration tool validate again to ensure that the error(s) has\have been corrected. If the error(s) still persists, please contact [Azure DevOps Services customer support](#).

ISVError:300005

IMPORTANT

Ensure that you have a backup of your collection and configuration databases before running the below commands to fix this error.

ISVError:300005 indicates that a non-group identity is a member of an everyone group, more commonly known as the Valid Users groups. Valid Users groups are created by default for all projects and collections. They're uneditable groups that only contain other Azure DevOps Server groups as members. In the case of ISVError:300005, a non Azure DevOps Server group identity, such as an AD group or user identity, has a direct membership in a Valid Users group.

Since Valid Users groups can't be edited directly or through TFSSecurity.exe, correcting the invalid membership will need to be done by running a SQL statement against the configuration database to remove the offending identity. Carefully examine the error message(s) the data migration tool highlighted. You will need copy down the GroupSid, MemberId, and Scopeld as these values will need to be placed into the templated command below.

```
DECLARE @p6 dbo.typ_GroupMembershipTable

INSERT into @p6 values('{GroupSid}', 'Microsoft.TeamFoundation.Identity', '{MemberId}', 0)

EXEC prc_UpdateGroupMembership
@partitionId=1,@scopeId='{ScopeId}',@idempotent=1,@incremental=1,@insertInactiveUpdates=0,@updates=@p6,@event
Author='9EE20697-5343-43FC-8FC5-3D5D455D21C5',@updateGroupAudit=0
```

Below is an example ISVError:300005 message from the data migration tool.

```
ISVError:300005 Unexpected non group identity was found to have direct membership to everyone group.
GroupSid:S-1-9-1551374245-3746625149-2333054533-2458719197-2313548623-0-0-0-3, MemberId:76050ddf-4fd8-48c4-
a1ff-859e44364519, ScopeId:7df650df-0f8b-4596-928d-13dd89e5f34f
```

Copy the GroupSid, MemberId, and Scopeld into the templated SQL command.

Run the completed command against the Azure DevOps Server configuration database. This will need to be repeated for each ISVError:300005 instance that the data migration tool found. Errors with the same scope ID can be batched into one command. Once the commands have been executed you will need to run the data migration tool validate again to ensure that the errors have been corrected. If the errors still persist, please contact [Azure DevOps Services customer support](#).

IMPORTANT

The collection that you're trying to fix the error for must be attached.

If you receive a -1 result from running the command, ensure that your collection database that produced the error is attached to your Azure DevOps Server instance and that you're running the command on the configuration database.

AAD timeout exception

On rare occasions some might receive an AAD timeout error when running the data migration tool prepare command.

Exception Message: Request failed (type AadGraphTimeoutException)

This error means that the requests to AAD to find the matching AAD identities for users in your collection timed out. Generally, this error can be resolved by waiting to run the prepare command at a less busy time of the day. Such as after regular business hours.

In the event that the error continues there are few troubleshooting steps which should be undertaken. First, you will want to test your connection to AAD from the machine running the prepare command. Follow the below steps and see if you can retrieve information on a user in your AAD.

Open PowerShell in elevated mode and add replace 'someone@somecompany.com' below with a user you expect to be present in your company's AAD.

```
//Install the AzureAD PowerShell module - ensuring to select Yes to All
Install-Module AzureAD

// Install the MSOnline PowerShell module - ensuring to select Yes to All
Install-Module MSOnline

// Connect to AAD and use your AAD credentials (someone@somecompany.com) to login when the pop-up appears
Connect-MsolService

// Try to retrieve information on a user from your AAD
Get-MsolUser -UserPrincipalName someone@somecompany.com
```

If any of the above steps fail or you're unable to look up a user's identity, that's a strong indication that there is a connection issue between the machine running the prepare command and AAD. You should run a network trace while executing the prepare command to ensure that nothing within your own network is stopping the calls from reaching AAD. If you've confirmed that the problem is not with your network then you will need to reach out to Azure support for assistance with troubleshooting.

If you are able to get information back on a user, open your log file from the prepare attempt and look for a line like the following.

```
Number of active users is {Number of Users}.
```

If this number is in the high five-digits or even six-digits ranges then it could be an indication that the volume of identities being mapped require more time than the timeout limit provides. You should inspect your collection for inclusions of large AD groups such as an 'everyone' group. If possible remove these groups and try again. If you still can't resolve this error then please reach out to [Azure DevOps Services customer support](#).

Resolve process errors

See the separate [Process Templates](#) page for details on resolving common process errors.

Resolve field validation errors

VS403442

In order to migrate successfully, you must rename field `{TFSfieldName}`. Given name `{TFSfieldName}` is reserved for field `{VSTSfieldName}`.

Sometimes your local collection may have a field whose name may conflict with Azure DevOps Services system field. To resolve this error, you must change name of your collection field. use `changefield` command from [witadmin](#)

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldName  
/name:newFieldName
```

VS403443

In order to migrate successfully, you must rename field `{TFSfieldName}` to `{VSTSfieldName}`. Given name for `{TFSfieldName}` is `{TFSfieldName}`

Sometimes your local collection may have different name for a particular field. To resolve this error, use `changefield` command from [witadmin](#)

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldName  
/name:VSTSfieldName
```

VS403444

In order to migrate successfully, you must set type of field `{TFSfieldName}` to `{Type}`. Given type for `{TFSfieldName}` is `{collectionType}`.

Sometimes your local collection may have different type for a particular field. Presently [witadmin](#) allows type change for only those fields which are either of HTML or PlainText type. If your field type is either HTML or PlainText, then you can change its type to required type using [witadmin](#).

```
witadmin changefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldName  
/type:PlainText | HTML
```

NOTE

If your field type is something different than HTML|PlainText and field data is not important or field is not being used in any project, then we recommend using [witadmin](#) to delete that field.

```
witadmin deletefield /collection:http://AdventureWorksServer:8080/DefaultCollection /n:TFSfieldName
```

IMPORTANT

Using [witadmin](#) to delete a field will result in loss of field data across the collection.

Resolve import errors

Hit a failure when running your import? Failures in the import space fall into one of two categories. Verification failures happen when the import fails to start. The indication that this has occurred is when the data migration tool attempts to queue an import, but returns an error instead. Import failures happen when the import was queued successfully in the data migration tool, but failed after that point. The individual that queued the import will receive a failure email if this happens.

Verification failures

Verification failures happen when the import fails to start. Issues falling into this category mean that something with your import request isn't valid. Look up your error message below and follow the recommended guidance on how to resolve the error. After that your team can try to queue the import again.

VS403254

```
VS403254: Region {0} may not be used for the Import, it is not a supported region.
```

The region that you entered for your Azure DevOps Services import isn't supported. Open your import specification file and update the region that you've provided with the correct short name for the [region](#) you want to import into.

VS403249

```
VS403249: The organization {0} already exists. Please select a different name and try the import again.
```

All Azure DevOps Services imports go into a new organization that is created at import time. This error indicates that the organization name your team has selected is already being used by an existing organization. Select a different name and update the import specification file before retrying the import.

VS403250 & VS403286

```
VS403250: The dacpac is not a detached Azure DevOps Server Collection database.
```

```
VS403286: The dacpac is from a Azure DevOps Server Configuration database. You must use a detached Azure DevOps Server Collection database.
```

The DACPAC is not built off a detached collection. The collection database will need to be [detached](#) and the DACPAC generated again.

VS403243

```
VS403243: Unable to connect to the database using the provided SQL Connection String {0}.
```

Unable to make a connection to the database using the provided SQL Connection String. Review the parameters that were provided to ensure they're correct and try again.

VS403260 & VS403351

VS403260: The database is not detached.
VS403351: The DACPAC or source database is missing an expected table. It's possible that the database was not correctly detached from Azure DevOps Server.

The database is not detached. It will need to be [detached](#) and the import queued again.

VS403261

VS403261: The SQL connection string must use encryption.

The connection string must be encrypted otherwise the password will be sent in the clear. Please add "Encrypt=true" to your SQL connection string.

VS403262

VS403262: The SQL connection string must use SQL Authentication, Integrated Authentication is not supported.

Please add "Integrated Security=False" to your SQL connection string.

VS403263

VS403263: The User ID {0} must be member of the database role {1}.

This error means that your SQL login user does not have the required database role. Please make sure '[TFSEXECROLE](#)' is assigned to the login.

There is a known issue with using sp_addrolemember to add 'TFSEXECROLE' to an existing SQL login. The role membership is not applied until all open connections using that identity are closed. If you're hitting the above error and have confirmed your identity has this role, it's recommended that you create a new identity for your import. Details on how to create a new SQL login that's ready to be used for import can be found at <https://aka.ms/AzureDevOpsImportLargeCollection>.

VS403264

VS403264: The database is not a Azure DevOps Server Collection database, it cannot be used for import.

The connection string does not point to a Azure DevOps Server Collection database.

VS40325

VS403255: The collection cannot be imported due to an ongoing post upgrade job. Please wait and try again later

The Azure DevOps Server Update has queued the file migration job. Imports cannot be performed until this job has completed. The completion time for this job is dependent on the size of the collection. Job progress can be tracked by running the below query on the collection database:

```
SELECT COUNT (*) as remaining_files_to_migrate
FROM   tbl_FileReference
WHERE  PartitionId > 0
       AND MigrateFileDialog IS NOT NULL
```

Once the number of files remaining to migrate is zero you can run the data migration tool.

VS403282

VS403282: The source location parameter contains a new line character. Please ensure the SAS key is defined on a single line in the import specification file.

There is a new line character in the source location value, this could have been left over when copying the SAS key from your windows console, please remove the line break and try again.

VS403271

VS403271: It appears that your DACPAC was uploaded to East US. It's required that customers targeting Central US for import put their DACPACs in Central US. Please move your DACPAC to Central US and requeue the import.

Your import files and DACPAC are not located in the **required** Azure region to complete the import to your target Azure DevOps Services region. Please [Create a new windows azure storage account](#) in the required region and copy your files. Below is an example of how to copy your data using AzCopy.

```
AzCopy.exe /Source:https://accountSCUS.blob.core.windows.net/mycontainer /SourceKey:"primary access key"  
/Dest:https://accountCUS.blob.core.windows.net/mycontainer /DestKey:"primary access key" /S
```

VS403316

VS403316: An inconsistency was detected in some TFVC files for this collection. The inconsistency needs to be corrected prior to running an import to Azure DevOps Services. Please reach out to <https://aka.ms/AzureDevOpsImportSupport> for assistance with addressing this issue.

An inconsistency was detected in some TFVC files within your collection. To resolve the error you will need to work Azure DevOps Services [customer support](#). Please open a support ticket and they will assist you with correcting the error.

VS403366

VS403366: A problem occurred while attempting to connect to your database. Please verify that your connection string is correct and that all required IP addresses for Azure DevOps Services have been provided exceptions for your machines firewall.

List of Azure DevOps Services IPs:

The data migration tool was unable make a connection to the SQL Azure VM. Verify that you've entered the information correctly in your connection string and that you can connect to the VM. The IPs that the error message lists are for Azure DevOps Services. Azure DevOps Services IPs can change temporarily during deployments. Please add them to your firewall exceptions and try queuing the import again.

VS403373

Importing multiple copies of the **SAME** collection is not supported by the data migration tool for Azure DevOps. However, we **DO** support importing **split** copies of a collection. What you need to do is change the GUID for the *DataImportCollectionID*.

From SQL Server Management Studio (SSMS) open the extended properties for the split copies that haven't been imported yet. Add a newly generated GUID to the "TFS_DATAIMPORT_COLLECTIONID" property. Then re-run the prepare command and use the new import.json to queue the import.

Import failures

When an import fails, the individual that queued the import will receive an email notification. Most of the time

this email will include a reason for the failure. If it does, use the troubleshooting steps provided in the email and this page to resolve the errors and try your import again.

If the error is more complex then the email will provide instructions on how to file a customer [support case](#). After submitting a customer support case, your team will need to roll back by bringing your Azure DevOps Server instance back online and reattach your collection. This will allow your team members to continue working. It's recommended not to attempt the import again until the issue causing the failure has been resolved.

Learn how to add continuous security validation to your CI/CD pipeline

10/2/2019 • 9 minutes to read • [Edit Online](#)

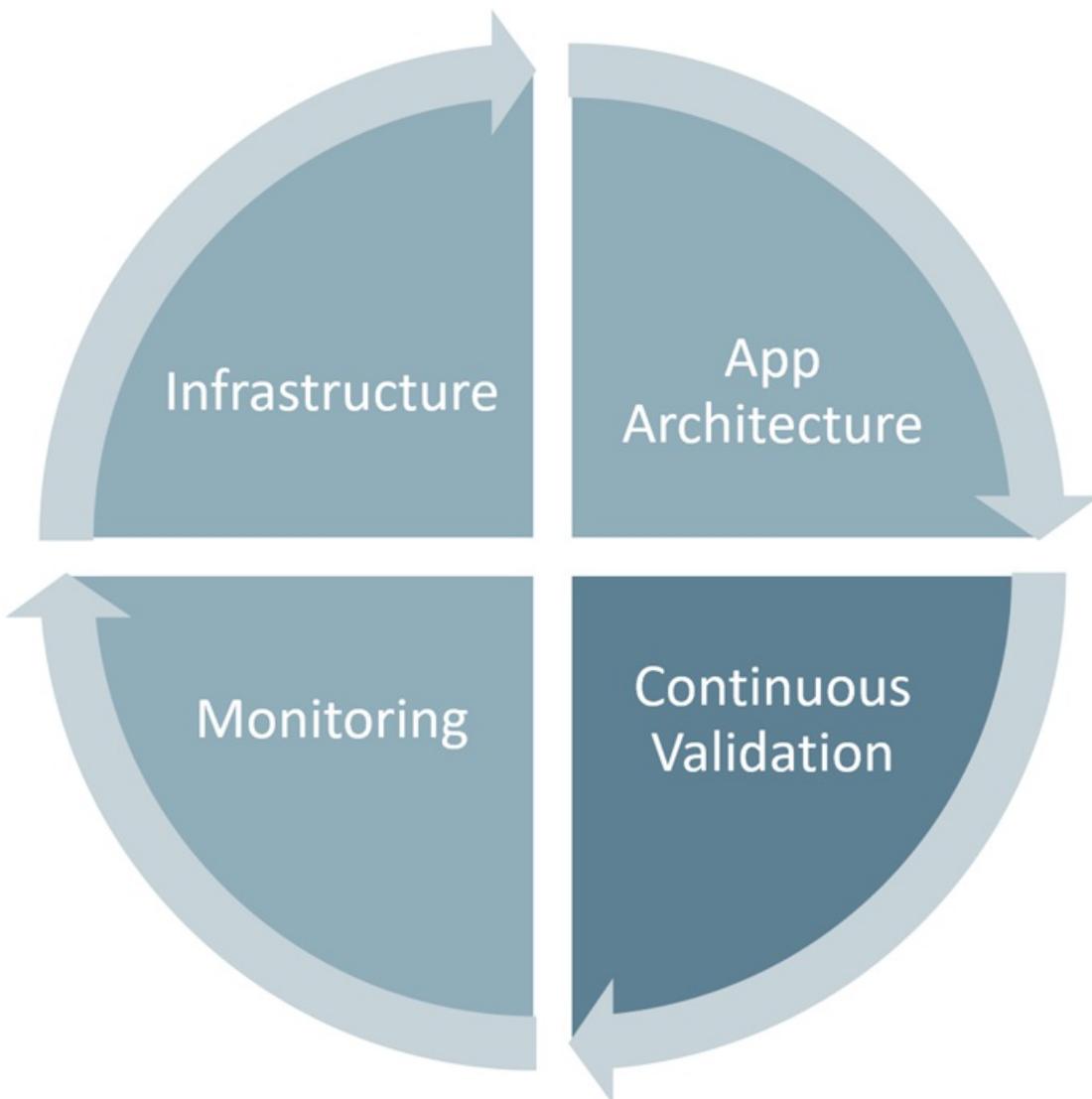
Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015

Are you planning Azure DevOps continuous integration and deployment pipelines? You probably have a few questions, such as:

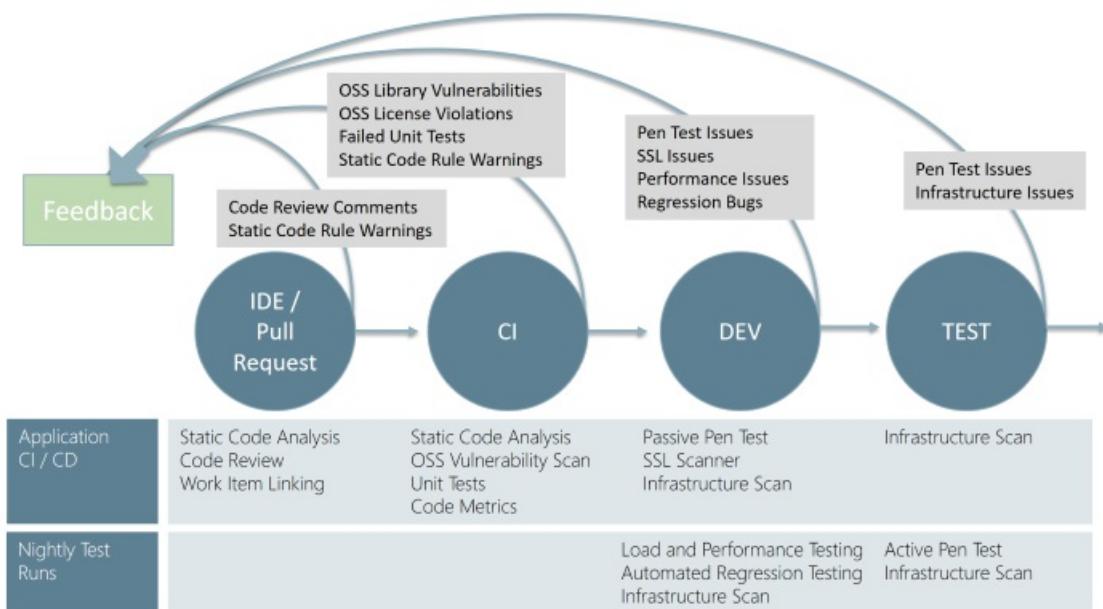
- How do you ensure your application is safe?
- How do you add continuous security validation to your CI/CD pipeline?

DevOps practices are allowing businesses to stay ahead of the competition by delivering new features faster than ever before. As the frequency of production deployments increases, this business agility cannot come at the expense of security. With continuous delivery, how do you ensure your applications are secure and stay secure? How can you find and fix security issues early in the process? This begins with practices commonly referred to as DevSecOps. DevSecOps incorporates the security team and their capabilities into your DevOps practices making security a responsibility of everyone on the team. This article will walk you through how to help ensure your application is secure by adding continuous security validation to your CI/CD pipeline.

Security needs to shift from an afterthought to being evaluated at every step of the process. Securing applications is a continuous process that encompasses secure infrastructure, designing an architecture with layered security, continuous security validation, and monitoring for attacks.



Continuous security validation should be added at each step from development through production to help ensure the application is always secure. The goal of this approach is to switch the conversation with the security team from approving each release to approving the CI/CD process and having the ability to monitor and audit the process at any time. When building greenfield applications, the diagram below highlights the key validation points in the CI/CD pipeline. Depending on your platform and where your application is at in its lifecycle, you may need to consider implementing the tools gradually. Especially if your product is mature and you haven't previously run any security validation against your site or application.



IDE / Pull Request

Validation in the CI/CD begins before the developer commits his or her code. Static code analysis tools in the IDE provide the first line of defense to help ensure that security vulnerabilities are not introduced into the CI/CD process. The process for committing code into a central repository should have controls to help prevent security vulnerabilities from being introduced. Using Git source control in Azure DevOps with branch policies provides a gated commit experience that can provide this validation. By enabling [branch policies](#) on the shared branch, a pull request is required to initiate the merge process and ensure that all defined controls are being executed. The pull request should require a code review, which is the one manual but important check for identifying new issues being introduced into your code. Along with this manual check, commits should be linked to work items for auditing why the code change was made and require a continuous integration (CI) build process to succeed before the push can be completed.

CI (Continuous Integration)

The CI build should be executed as part of the pull request (PR-CI) process discussed above and once the merge is complete. Typically, the primary difference between the two runs is that the PR-CI process doesn't need to do any of the packaging/staging that is done in the CI build. These CI builds should run static code analysis tests to ensure that the code is following all rules for both maintenance and security. Several tools can be used for this.

- [Visual Studio Code Analysis and the Roslyn Security Analyzers](#)
- [Checkmarx](#) - A Static Application Security Testing (SAST) tool
- [BinSkim](#) - A binary static analysis tool that provides security and correctness results for Windows portable executables
- [Other 3rd party tools](#)

Many of the tools seamlessly integrate into the Azure Pipelines build process. Visit the VSTS Marketplace for more information on the integration capabilities of these tools.

In addition to code quality being verified with the CI build, two other tedious or ignored validations are scanning 3rd party packages for vulnerabilities and OSS license usage. Often when we ask about 3rd party package vulnerabilities and the licenses, the response is fear or uncertainty. Those organizations that are trying to manage 3rd party packages vulnerabilities and/or OSS licenses, explain that their process for doing so is tedious and manual. Fortunately, there are a couple of tools by [WhiteSource Software](#) that can make this identification process almost instantaneous. The tool runs through each build and reports all of the vulnerabilities and the licenses of the 3rd party packages. WhiteSource Bolt is a new option, which includes a 6-month license with your Visual Studio Subscription. Bolt provides a report of these items but doesn't include the advanced management and alerting capabilities that the full product offers. With new vulnerabilities being regularly discovered, your build reports could change even though your code doesn't. Checkmarx includes a similar WhiteSource Bolt integration so there could be some overlap between the two tools. See, [Manage your open source usage and security as reported by your CI/CD pipeline](#) for more information about WhiteSource and the Azure Pipelines integration.

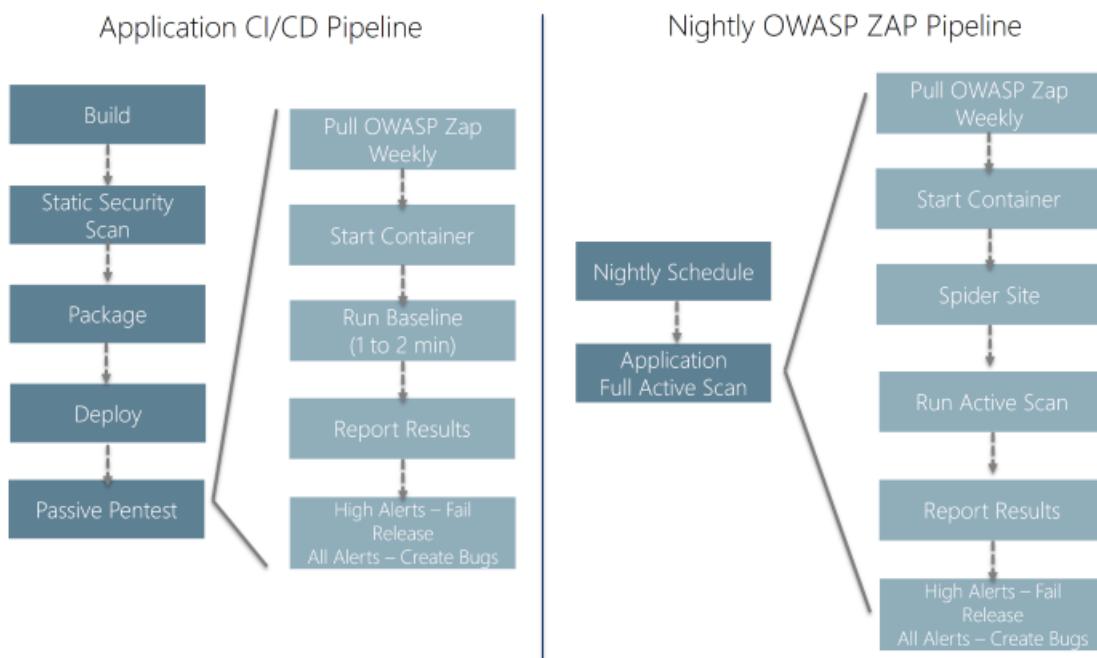
Application Deployment to DEV and TEST

Once your code quality is verified, and the application is deployed to a lower environment like development or QA, the process should verify that there are not any security vulnerabilities in the running application. This can be accomplished by executing automated penetration test against the running application to scan it for vulnerabilities. There are different levels of tests that are categorized into passive tests and active tests. Passive tests scan the target site as is but don't try to manipulate the requests to expose additional vulnerabilities. These can run fast and are usually a good candidate for a CI process that you want to complete in a few minutes. Whereas the Active Scan can be used to simulate many techniques that hackers commonly use to attack websites. These tests can also be referred to dynamic or fuzz tests because the tests are often trying a large number of different combinations to see

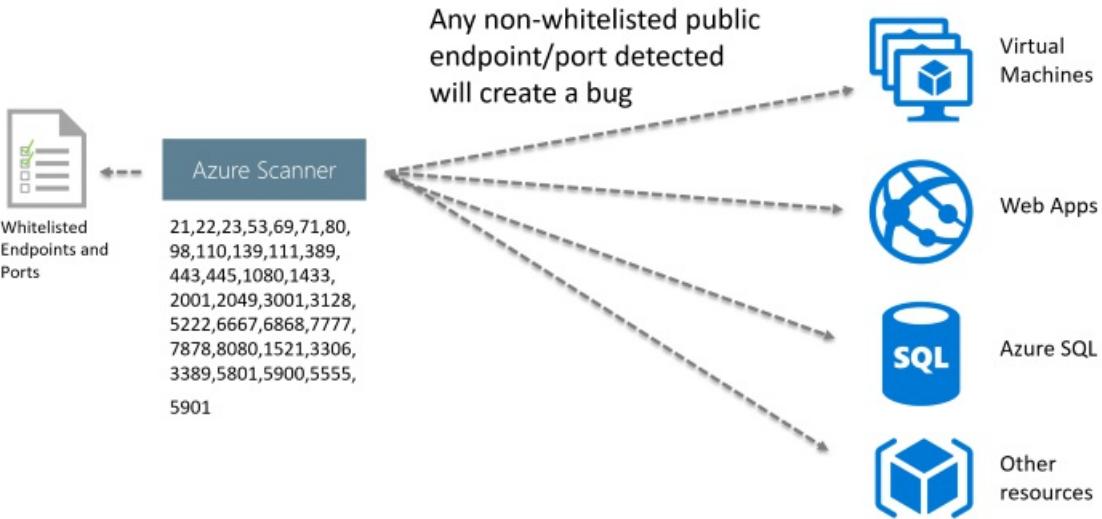
how the site reacts to verify that it doesn't reveal any information. These tests can run for much longer, and typically you don't want to cut these off at any particular time. These are better executed nightly as part of a separate Azure DevOps release.

One tool to consider for penetration testing is OWASP ZAP. [OWASP](#) is a worldwide not-for-profit organization dedicated to helping improve the quality of software. ZAP is a free penetration testing tool for beginners to professionals. ZAP includes an API and a weekly docker container image that can be integrated into your deployment process. The detailed how-to steps are outside the scope of this article. Refer to the [OWASP ZAP VSTS extension](#) repo for details on how to set up the integration. Here we're going to explain the benefits of including this into your process.

The application CI/CD pipeline should run within a few minutes, so you don't want to include any long-running processes. The baseline scan is designed to identify vulnerabilities within a couple of minutes making it a good option for the application CI/CD pipeline. The Nightly OWASP ZAP can spider the website and run the full Active Scan to evaluate the most combinations of possible vulnerabilities. OWASP ZAP can be installed on any machine in your network, but we like to use the OWASP Zap/Weekly docker container within Azure Container Services. This allows for the latest updates to the image and also allows being able to spin up multiple instances of the image so several applications within an enterprise can be scanned at the same time. The following figure outlines the steps for both the Application CI/CD pipeline and the longer running Nightly OWASP ZAP pipeline.



In addition to validating the application, the infrastructure should also be validated to check for any vulnerabilities. When using the public cloud such as Azure, deploying the application and shared infrastructure is easy, so it is important to validate that everything has been done securely. Azure includes many tools to help report and prevent these vulnerabilities including Security Center and Azure Policies. Also, we have set up a scanner that can ensure any public endpoints and ports have been added to an allow list or else it will raise an infrastructure issue. This is run as part of the Network pipeline to provide immediate verification, but it also needs to be executed each night to ensure that there aren't any resources publicly exposed that should not be.



Once the scans have completed, the Azure Pipelines release is updated with a report that includes the results and bugs are created in the team's backlog. Resolved bugs will close if the vulnerability has been fixed and move back into in-progress if the vulnerability still exists.

The benefit of using this is that the vulnerabilities are created as bugs that provide actionable work that can be tracked and measured. False positives can be suppressed using OWASP ZAP's context file, so only vulnerabilities that are true vulnerabilities are surfaced.

Total tests	Pass percentage	Run duration
6	0%	0s

Test

- 0/6 Passed - OWASP ZAP Security Tests
- Incomplete or No Cache-control and Pragma HTTP Header Set
- Cookie No HttpOnly Flag
- Cookie Without Secure Flag
- Web Browser XSS Protection Not Enabled
- X-Content-Type-Options Header Missing
- X-Frame-Options Header Not Set

Even with continuous security validation running against every change to help ensure new vulnerabilities are not introduced, hackers are continuously changing their approaches, and new vulnerabilities are being discovered. Good monitoring tools allow you to help detect, prevent, and remediate issues discovered while your application is running in production. Azure provides a number of tools that provide detection, prevention, and alerting using rules such as OWASP Top 10 / modSecurity and now even using machine learning to detect anomalies and unusual behavior to help identify attackers.

Minimize security vulnerabilities by taking a holistic and layered approach to security including secure infrastructure, application architecture, continuous validation, and monitoring. DevSecOps practices enable your entire team to incorporate these security capabilities throughout the entire lifecycle of your application. Establishing continuous security validation into your CI/CD pipeline can allow your application to stay secure while you are improving the deployment frequency to meet needs of your business to stay ahead of the competition.

Reference information

- [BinSkim](#) - A binary static analysis tool that provides security and correctness results for Windows portable executables
- [Checkmarx](#) - A Static Application Security Testing (SAST) tool
- [Manage your open source usage and security as reported by your CI/CD pipeline](#)
- [OWASP](#)
- [OWASP ZAP VSTS extension](#)
- [WhiteSource Software](#)
- [Visual Studio Code Analysis and the Roslyn Security Analyzers](#)

Authors: Mike Douglas | Find the origin of this article and connect with the ALM | DevOps Rangers [here](#)

(c) 2017 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Build and Deployment Automation Case Study for World Wide Time Keeping: Higher Quality and Faster Delivery in an Increasingly Agile World

10/2/2019 • 8 minutes to read • [Edit Online](#)

Author: Vaibhav Rajeev Thombre

October 2015

In an Agile world, delivering quick and frequent releases for large, complex systems with multiple components becomes cumbersome and time-consuming if done manually, because each component has a high degree of complexity and requires a lot of resource intervention and configuration to ensure that it works as expected.

That's why many teams opt for Build and Deployment Automation to ensure faster releases and reduce manual intervention. However, automating multiple components of a system has its own challenges. Even though releases can be automated in silos, if we need a one-click deployment for the entire system, we need to have an automation framework that can automate an entire custom workflow.

Throughout this paper, we give insight on our project - World Wide Time Keeping - and how we implemented build and deployment automation using Gated Check-ins, Code Analysis, and Fortify Integrations. We discuss build and deployment automation by using PowerShell scripts and how we can create custom workflows and deploy all at once using Release Management. We also talk about how these can help you cut down your engineering cycle time and play an important role in hitting Production Ready at Code Complete (PRCC) goals. This lets you have a Continuous Integration Continuous Delivery (CICD) Project and helps you go faster, without introducing issues.

This content is useful for SWE teams who are working in an Agile model for large, complex systems and want to cut down their release cycles and deliver faster. We assume that readers have a fundamental knowledge of Engineering Cycles and their phases (Develop/Test/Build/Deploy) and a fundamental knowledge of Agile practices and delivery cycles.

Build automation

Many teams have multiple requirements for build, but the following practices can be applied to most teams. You may adopt the whole approach or just implement the components that work out best for you.

Daily Builds: Have a build pipeline for scheduled builds. Aim for a daily schedule with builds released to the internal SWE environment by the end of each day.

One-click builds for non-internal environments: For Integration/UAT environments, you automate the builds. Instead of scheduling them on a per day basis, you can trigger them by queuing them in VSTF. (The reason for not scheduling them is that a build is not required on Integration/UAT environments on a daily basis. Rather, they tend to happen on an as-needed basis. This will depend on your team's needs and you can adopt the rhythm that works best for your team.)

Gated Check-ins: Set up gated check-ins to ensure that only code that complies and passed unit testing gets checked in. It ensures that code quality remains high and that there are no broken builds. Integrate Fortify and Code Analysis to get further insight into code quality.

Code Analysis Integrations: To get insight into whether the code is of good quality or if any changes need to be

made, integrate Code Analysis into the build pipelines and set the threshold to low. The changes can be identified and fixed early, which is required in the Agile world.

Fortify Integrations: Use Fortify for security-based checks of the build pipelines associated with your check-ins and daily builds. This ensures that any security vulnerabilities are identified as soon as possible and can be fixed quickly.

Deployment automation

Use deployment scripts

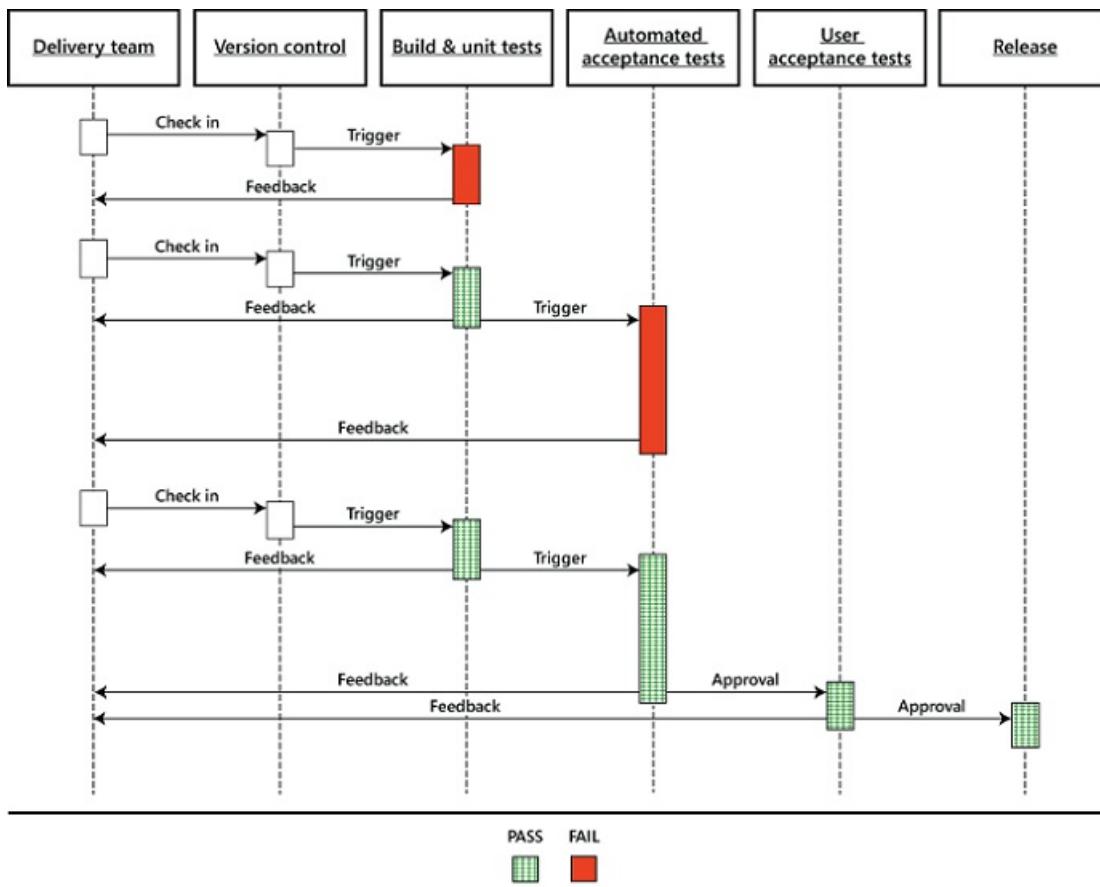
Deployments for internal SWE environment: Set up the internal SWE environments deployments with the daily automated builds by integrating the build pipelines with the deployment scripts. All the checked-in changes will then be deployed at the end of each day, without any manual intervention.

This way, the latest build is present in the SWE environment in case you would like to demo the product to stakeholders.

Deployments for Integration/UAT environments: For Integration/UAT environments, you can integrate the scripts with the build pipelines without scheduling them and trigger them on an as-needed basis. Because you have set up one-click builds for them, when the build completes successfully, the scripts get executed at the end and the product is deployed. Therefore, you do not have to manually deploy the system. Instead it's deployed automatically by simply queuing a build.

The release pipeline

In theory, a release pipeline is a process that dictates how you deliver software to your end users. In practice, a release pipeline is an implementation of that pattern. The pipeline begins with code in version control and ends with code deployed to the production environment. In between, a lot can happen. Code is compiled, environments are configured, many types of tests run, and finally, the code is considered "done". By done, we mean that the code is in production. Anything you successfully put through the release pipeline should be something you would give to your customers. Here is a diagram based on the one you will see on Jez Humble's [Continuous Delivery](#) website. It is an example of what can occur as code moves through a release pipeline.



Use Release Management

If your team is working on Azure-based components - web apps, services, web jobs, and so on - you can use Release Management for automating deployments.

Release Management consists of various pre-created components which you can configure and use either independently or in conjunction with other components through workflows.

You might face pain points when you manually deploy an entire system. For a large complex system with multiple components, like service, web jobs, and dacpac scripts, here are example pain points:

- A large amount of time goes into configuration of each component
- Deployment needs to be done separately for each, adding to the overall deployment time.
- Multiple resources have to be engaged to ensure that the deployments happen as expected.

How Release Management (RM) solves them:

- RM allows you to create custom workflows which sequence the deployment to ensure that the components get deployed as soon as their dependencies have been deployed.
- Configurations can be stored in RM to ensure that configuration per deployment is not required.
- It automates the entire workflow which ensures manual intervention is not required and resources can be utilized for functional tasks.

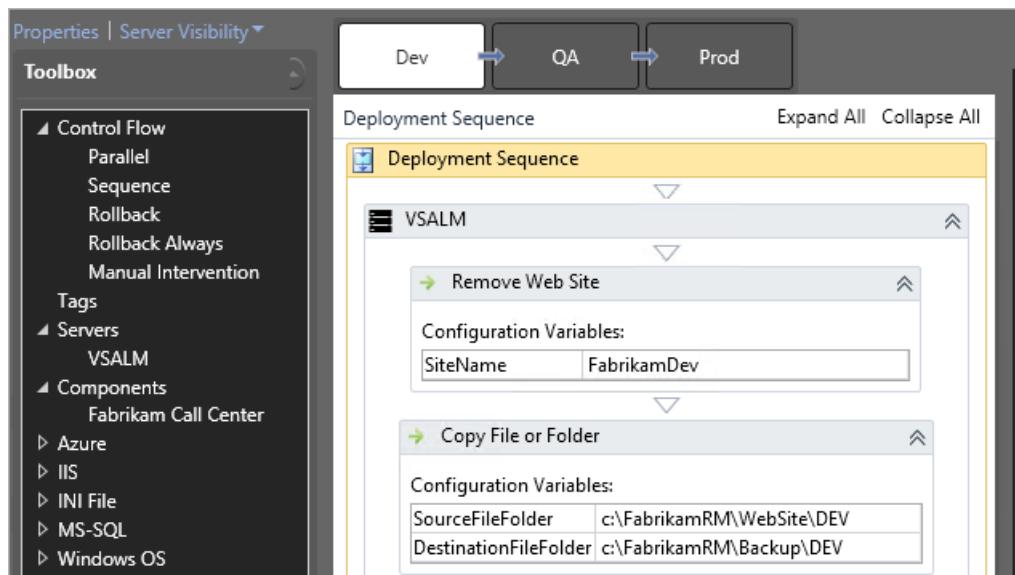
Key takeaways

- Set up Automated Builds scheduled for the rhythm that works best for your product and Implement Gated Check-ins.
- Integrate Code Analysis and Fortify into the build setup to improve the code quality and security of the application

- Set up daily automated deployments to the internal SWE environments and set up one click deployments to environments like UAT and Prod.
- Use Release Management to set up custom workflows for your releases and triggering them with a single click.

To use Release Management, you need to set up the following components:

- **RM Server:** Is the central repository for configuration and release information.
- **Build Agent:** This is a machine (physical or VM) that you set up at your end on which you will run all your builds and deployments.
- **Environments:** This signifies the environment which will be used in conjunction with your machine that you have set up.
- **Release Paths:** You need to create Release Paths for the multiple releases that you want to automate for multiple environments - internal SWE envs, INT, UAT, and so on.
- **Build Components:** The build component is used to configure the build and change any environment specific configurations. It picks up the build from the remote machine in which VSTF auto-generates the builds as per the build pipeline and runs the configuration changes that are defined within it.
- **Release Templates:** Release template defines the workflow that you have set up as per your specific needs of deployment. It also defines the sequence in which the RM components are to get executed. You need to integrate your build pipeline from Team Foundation Server (TFS) with the release template to enable continuous delivery. You can either pick up the latest build or select the build.



Conclusion

In this paper, we discussed the various engineering practices we can use for enabling faster product delivery with higher quality. We discussed:

- **Build Automation:** Builds can be set up for triggering on a schedule or on an ad-hoc basis just by a single click. It can vary based on the rhythm that works best for your team. Gated check-ins should be set up on top of the build pipelines to accept only the check-ins which meet the criteria bar.
- **Code Analysis and Fortify Integration:** The build pipelines should be integrated with Code Analysis and Fortify to trigger on a schedule and also with the Gated Check-ins. Code Analysis will improve the code quality and Fortify will point out the security-based gaps in the application, if any.
- **Deployment Automation:** You can integrate PowerShell scripts with your build pipelines to achieve

deployment automation. You can also use Release Management to set up custom workflows and integrate it with your TFS to pick up the latest builds or even select builds.

We also discussed the benefits that we found by taking up these practices:

- Minimal wastage of time due to automations of build, deploy phases
- Higher code quality due to Gated check-ins (with integrated Test Automation), Code Analysis, and Fortify Integration
- Faster delivery
- Will enable you to hit Production Ready at Code Complete (PRCC)
- Will enable you to hit Continuous Integration & Continuous Delivery targets (CI/CD)

References

[1] Visual Studio team, [Automate deployments with Release Management](#), MSDN Article

[2] Visual Studio team, [Build and Deploy Continuously](#), MSDN Article

[3] Visual Studio team, [Building a Release Pipeline with Team Foundation Server 2012](#), MSDN Article

(c) 2015 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Explore how to progressively expose your features in production for some or all users

10/2/2019 • 6 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In today's fast-paced, feature-driven markets, it's important to continuously deliver value and receive feedback on features quickly and continuously. Partnering with end users to get early versions of features vetted out is valuable.

Are you planning to continuously integrate features into your application while they're under development? You probably have a few questions, such as:

- How can you toggle features to hide, disable, or enable features at run-time?
- How can you revert a change deployed to production without rolling back your release?
- How can you present users with variants of a feature, to determine which one performs better?

This topic aims to answer these questions and share an implementation of feature flags (FF) and A|B testing used with Azure DevOps extensions.

Considerations

Before you introduce feature flags to your engineering process, it's important to consider:

- Which users are you planning to target? For example, do you want to target specific or all users?
- Would you like users to decide which features they want to use?
- What's the value of embracing feature flags as part of your engineering process?
- What's the cost to implement feature flags in your engineering process?

Before you flip your first feature flag in production, take the time to read:

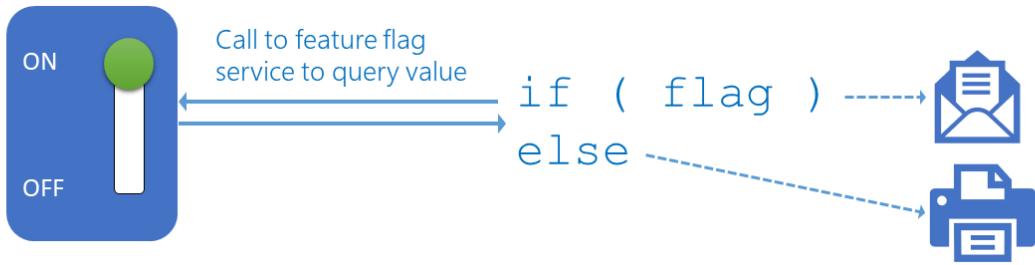
- "[A Rough Patch](#)", by Brian Harry
- "[Feature Flags with Branching](#)", by LaunchDarkly

What are Feature Flags (FF)?

NOTE

A feature flag is also known as a feature toggle, feature switch, feature flipper, or conditional feature. They were popularized by [Martin Fowler](#).

Feature flags support a customer-first DevOps mindset, to enable (expose) and disable (hide) features in a solution, even before they are complete and ready for release.



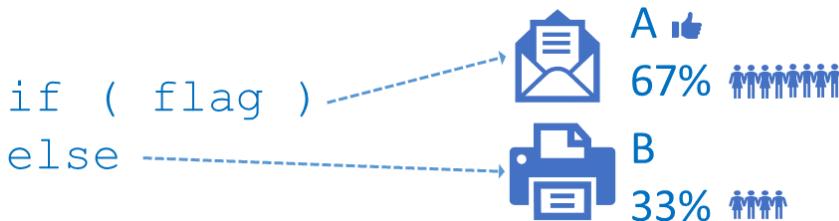
View a feature flag as an ON | OFF switch for a specific feature. As shown, you can deploy a solution to production that includes both an email and a print feature. If the feature flag is set (ON), you'll email, else you'll print.

When you combine a feature flag with an experiment, led by a hypothesis, you introduce A|B testing. For example, you could run an experiment to determine if the email (A) or the print (B) feature will result in a higher user satisfaction.

NOTE

A|B testing is also known as Split Testing. It's based on a hypothesis that's defined as:

For {user} who {action} the {solution} is a {how} that {value} unlike {competition} we {do better}



As shown, the email feature (option A) is more popular with your users and wins.

Evaluating Feature Flag solutions

As outlined in [how to implement feature flags and A|B testing](#), the ALM | DevOps Rangers evaluated a number of FF frameworks and solutions.

They chose the [LaunchDarkly](#) solution for several reasons:

- It's a "software as a service" (SaaS) solution
 - No custom solution to maintain
 - No upgrades - you're always using the latest and greatest
 - No servers - [LaunchDarkly](#) takes care of the machines that LaunchDarkly runs on
 - Always on and optimized for the Internet
- It's integrated with Azure DevOps Services and Team Foundation Server (TFS)
- It's simple and cost-effective for an open-source project

Common scenarios

You have a [CI/CD pipeline](#) for every Azure DevOps extension you're hosting on the [marketplace](#). You are using a ring deployment model and manual release approval checkpoints. The checkpoints are manual and time consuming, but necessary to minimize the chance of breaking the early-adopter and production user environments, forcing an expensive roll-back. You're looking for an engineering process, which enables you to:

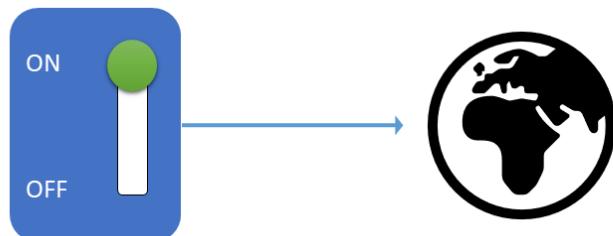
- Continuously deploy to production

- Never roll back in production
- Fine-tune the user experience in production

You have probably guessed it - feature flags!

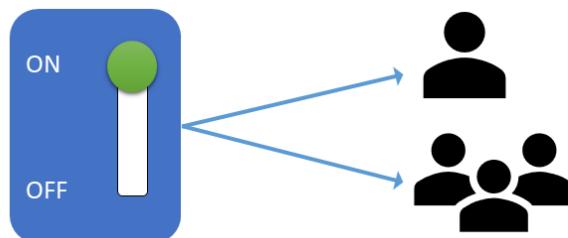
Enable or disable a feature for everyone

You would like to include hidden features in your release and enable them for **all** users in production. For example, you want to be able to collect verbose logging data for troubleshooting. Using a feature flag, you can enable and disable verbose logging as needed.



Enable or disable a feature for selected users

With this scenario, you can target specific users or groups of users. For example, you could enable the verbose logging feature for a specific user experiencing a problem or enable a preview feature for early adopters.

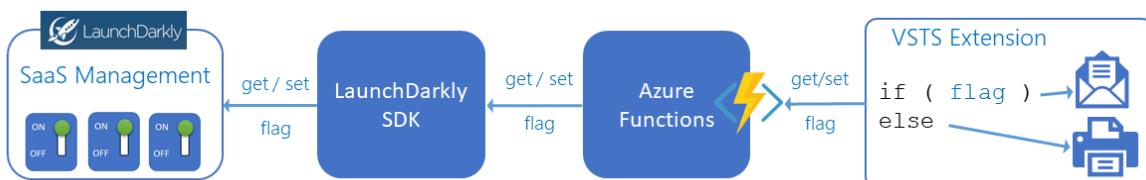


Enable | disable a feature as selected by user

Lastly, you'd like to give the users a list of preview features and allow each user to decide which feature to enable when. This scenario is key for feature validation, A|B testing, and giving the user flexibility and choice.

Manage features with feature flags in your engineering process

To protect the flags from malicious users, you need to generate and pass the hash of the user key to the LaunchDarkly API calls. As Azure DevOps extensions can only use client-side code, the ALM | DevOps Rangers chose Azure Functions to help generate the hash, as shown. Read [how we checked and fixed the 503 error and Performance issue in our Azure Function](#) for details.



Administration of feature flags is straight-forward.

1. You have a different environment for each extension, allowing you to have different feature flag values for Early Adopters and Users.
2. Optionally target specific users
3. Optionally target users that match custom rules
4. You have a default for each feature flag

You have granular control of each feature flag.

The screenshot shows the LaunchDarkly interface for managing feature flags. A sidebar on the left lists various sections: Roll-up Board, Users, Feature flags (highlighted), Goals, Dev console, Audit log, Quickstart, Documentation, and Integrations. A blue circle labeled '1' is positioned above the 'Feature flags' section. The main area is titled 'Enable telemetry' with a sub-section 'enable-telemetry'. It shows a targeting switch set to 'ON'. Below it are sections for 'Prerequisites' (with a '+ Add prerequisites' button) and 'Targeting' (with two sub-sections: 'Target individual users' and 'Target users who match these rules', both with '+ Add user targets' buttons). A blue circle labeled '2' is over the 'Target individual users' section. A blue circle labeled '3' is over the 'Target users who match these rules' section. A blue circle labeled '4' is over the 'Default rule' section, which contains a dropdown menu set to 'SERVE false'. At the bottom, there are two notes: 'If targeting is off, serve false' and 'If LaunchDarkly is unreachable or if no off variation is set, serve false'.

What's the value?

You're able to:

- Decouple deployment of releases and exposure of features
- Make changes (enable|disable features) without redeployment
- Fine-tune a user's features and experience
- Enable a user to optionally select preview features
- Hide an incomplete or faulty feature

What's the cost?

Aside from the licensing and maintenance cost of a feature flag service, you're adding technical debt to your code:

- With a true or false feature flag, your doubling your code and test paths
- With multi-value feature flag, you'll add even more code and test paths
- You'll need to identify and remove stale feature flags
- Understand and test the implications of flipping a feature flag

TIP

To minimize the costs associated with the use of feature flags, keep feature flags short lived and prevent multiple feature flags from interfering with each other by affecting the same functionality.

Conclusion

Now that you've covered the concepts and considerations of feature flags, you should be confident to explore ways to improve your CI/CD pipelines. While feature flags come at a cost, having a game plan to manage exposed features at run-time is invaluable.

Q & A

How does the Azure DevOps team use feature flags?

Buck's [feature flags blog post](#) and the [presentation/article](#) are great sources to get an understanding of the custom-built feature flag system used with Team Foundation Server (TFS) and Azure DevOps Services.

How do the ALM | DevOps Rangers use feature flags?

The Rangers use the [LaunchDarkly](#) SaaS solution. You can find their learnings in this [blog series](#).

When should you remove feature flags?

As Buck states, "Many feature flags go away and the teams themselves take care of that." The feature teams decide when to go delete the feature flags. It can get unwieldy after a while, so there's some natural motivation to go clean it up.

Is there a dependency on deployment rings?

No, rings and feature flags are symbiotic. Read [Feature Flags or Rings](#) for details.

Reference information

- [CI/CD pipeline examples](#)
- [DevOps @ Microsoft](#)
- [How to implement feature flags and A|B testing](#)

Authors: Willy Schaub | Find the origin of this article and connect with the ALM | DevOps Rangers [here](#)

(c) 2017 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Explore how to progressively expose your Azure DevOps extension releases in production to validate, before impacting all users

10/2/2019 • 7 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

In today's fast-paced, feature-driven markets, it's important to continuously deliver value and receive feedback on features quickly and continuously. Partnering with end users to get early versions of features vetted out is valuable.

Are you planning to build and deploy Azure DevOps extensions to production? You probably have a few questions, such as:

- How do you embrace DevOps to deliver changes and value faster?
- How do you mitigate the risk of deploying to production?
- How do you automate the build and deployment?

This topic aims to answer these questions and share learnings using rings with Azure DevOps extensions. For an insight into the Microsoft guidelines, read [Configuring your release pipelines for safe deployments](#).

One or more rings to rule your deployments

Deployment rings were first discussed in [Jez Humble's book](#). They support the production-first DevOps mindset and limit impact on end users, while gradually deploying and validating changes in production. Impact (also called **blast radius**), is evaluated through observation, testing, analysis of telemetry, and user feedback.

Considerations

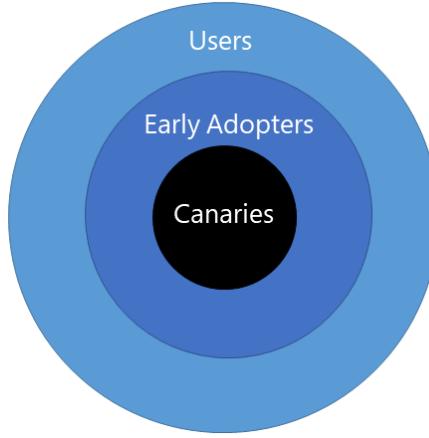
Before you convert your deployment infrastructure to a ringed deployment model, it's important to consider:

- Who are your primary types of users? For example, early adopters and users.
- What's your application topology?
- What's the value of embracing ringed deployment model?
- What's the cost to convert your current infrastructure to a ringed deployment model?

User types

In the shown example, users fall into three general buckets in production:

- **Canaries** who voluntarily test bleeding edge features as soon as they are available.
- **Early adopters** who voluntarily preview releases, considered more refined than the canary bits.
- **Users** who consume the products, after passing through canaries and early adopters.



NOTE

It's important to weigh out which users in your value chain are best suited for each of these buckets. Communicating the opportunity to provide feedback, as well as the risk levels at each tier, is critical to setting expectations and ensuring success.

Application topology

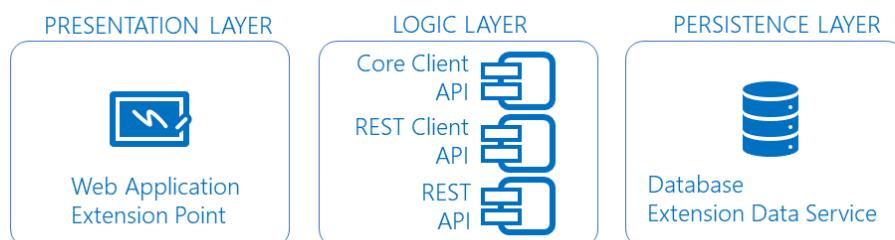
Next you need to map the topology of your application to the ringed deployment model. Limit the impact of change on end users and to continuously deliver value. Value includes both the value delivered to the end user and the value (return-on-investment) of converting your existing infrastructure.

NOTE

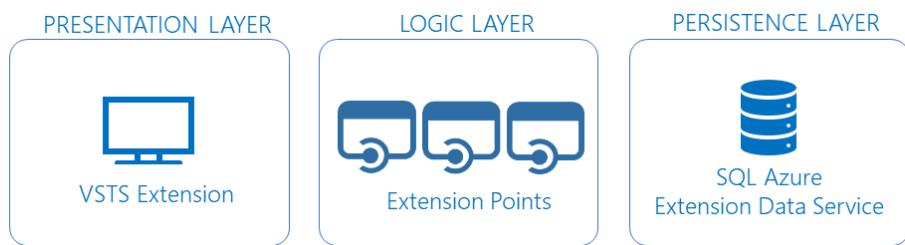
The ringed deployment model is not a silver bullet! Start small, prototype, and continuously compare impact, value, and cost.

At the application level, the composition of Azure DevOps extensions is innocuous, easy to digest, scale, and deploy independently. Each extension:

- Has one or more web and script files
- Interfaces with Core client
- Interfaces with REST client and REST APIs
- Persists state in cache or resilient storage



At the infrastructure level, the extensions are published to the [Visual Studio marketplace](#). Once installed in organization, they are hosted by the Azure DevOps service portal, with state persisted to Azure storage and/or the extension [data storage](#).



The extension topology is perfectly suited for the ring deployment model and to publish the extension to each deployment ring:

- A **private** development version for the canary ring
- A **private** preview version for the early adopter ring
- A **public** production version for the Users ring

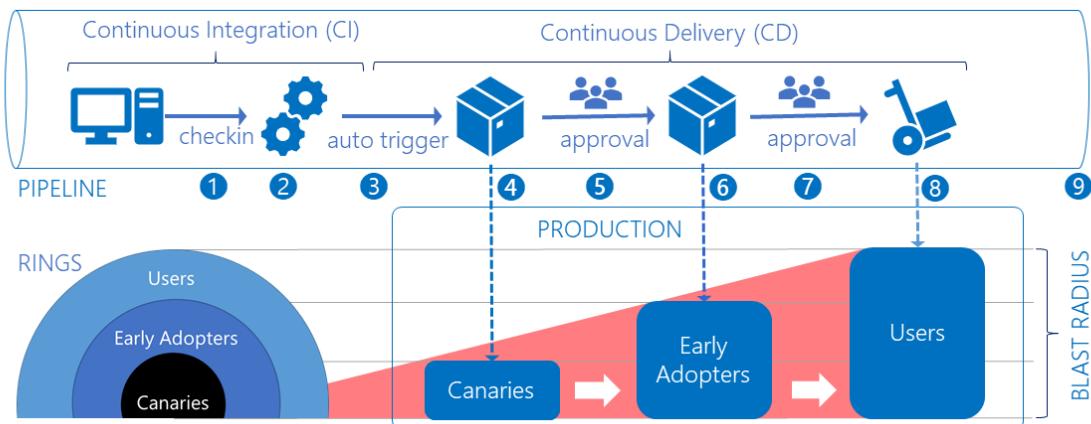
TIP

By publishing your extension as private, you're effectively limiting and controlling their exposure for users you explicitly invite.

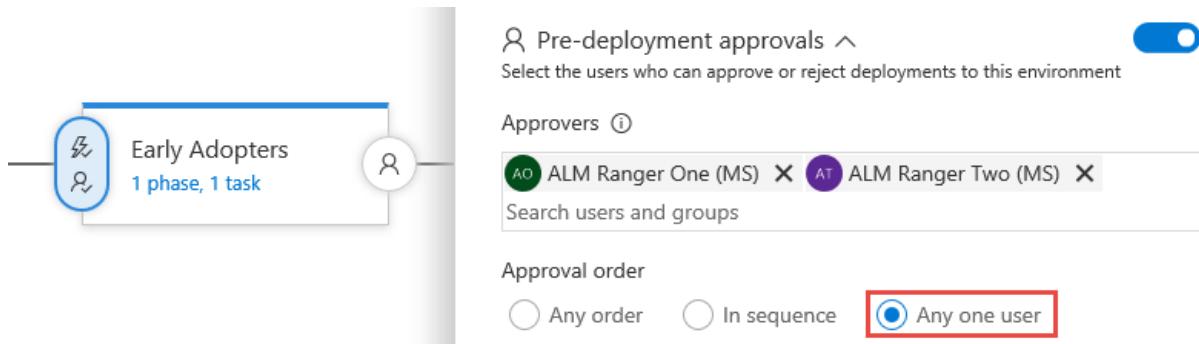
Moving changes through deployment rings

Let's observe how a change triggers and moves through the ring-based deployment process, using the [Azure DevOps Developer Tools Build Tasks](#) extension.

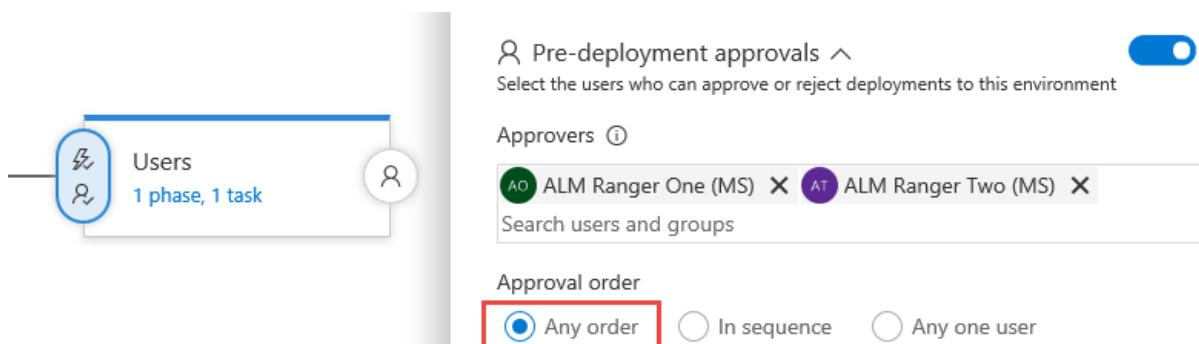
Azure DevOps Developer Tools Build Tasks extension is the secret sauce, used to package and publish Azure DevOps extensions to the Visual Studio Marketplace.



1. A developer from the [Countdown Widget extension](#) project commits a change to the [GitHub](#) repository.
2. The commit triggers a continuous integration build.
3. The new build triggers a continuous deployment trigger, which automatically starts the **Canaries** environment deployment.
4. The **Canaries** deployment publishes a private extension to the marketplace and shares it with predefined organizations. Only the **Canaries** are impacted by the change.
5. The **Canaries** deployment triggers the **Early Adopter** environment deployment. A pre-deployment approval gate requires any one of the authorized users to approve the release.



6. The **Early Adopter** deployment publishes a private extension to the marketplace and shares it with predefined organizations. Both the **Canaries** and **Early Adopter** are impacted by the change.
7. The **Early Adopter** deployment triggers the **Users** environment deployment. A stricter pre-deployment approval gate requires all of the authorized users to approve the release.



8. The **Users** deployment publishes a public extension to the marketplace. At this stage, everyone who has installed the extension in their organization is affected by the change.
9. It's key to realize that the impact ("blast radius") increases as your change moves through the rings. Exposing the change to the **Canaries** and the **Early Adopters**, is giving two opportunities to validate the change and hotfix critical bugs before a release to production.

NOTE

Review [CI/CD Pipelines](#) and [Approvals](#) for detailed documentation of pipelines and the approval features for releases.

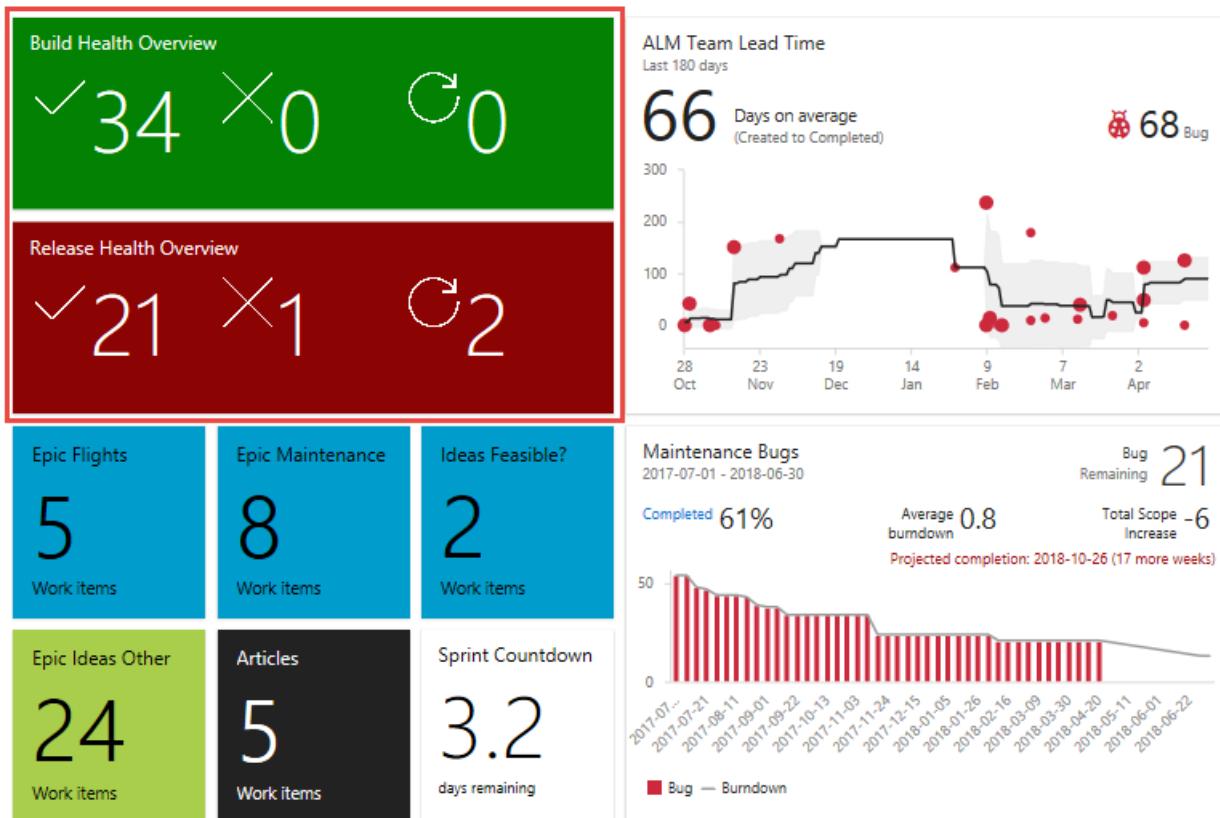
Dealing with monitoring and noise

You need **effective** monitoring and **actionable** alerts to detect and mitigate issues. Determine what type of data is important, for example infrastructure issues, violations, and feature usage. Focus on actionable alerts to avoid users ignoring them and missing high priority issues.

TIP

Start with high-level views of your data, visual dashboards that you can watch from afar, and drill-down as needed. Perform, regular housekeeping of your views and remove all noise. A visual dashboard tells a far better story than hundreds of notification emails, often filtered and forgotten by email rules.

Using the [Team Project Health](#) and out-of-the-box extensions you can build overview of your pipelines, lead and cycle times, and other information. In the sample dashboard, it's evident that there are 34 successful builds, 21 successful releases, 1 failed release, and 2 releases in progress.



What's the value?

Using a ring-deployment strategy you can gather feedback to validate your hypothesis. You can decommission old releases and distribute new releases without the risk of affecting all users.

Here's a summary of how the ALM | DevOps Ranger engineering process evolved with ring deployment models.

BEFORE USING RINGS		WITH RINGS
Manual and error prone	Build	Automated and consistent
Manual and error prone	Release	Automated and consistent
Hours	Time to build (TTB)	Seconds
Days	Time to release (TTR)	Minutes
Call from user	Issue detection	Proactive
Days to weeks	Issue resolution	Minutes to days

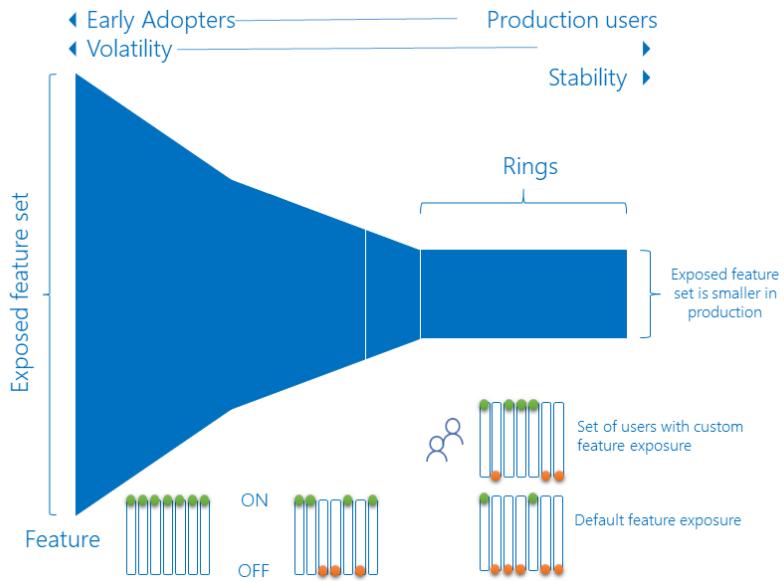
Key takeaways:

- Consistent and reliable automation
- Reduced response times
- Canaries experience the pain, not the users

Is there a dependency on feature flags?

No, rings and feature flags are symbiotic. Feature flags give you fine-grained control of features included in your

change. For example, if you're not fully confident about a feature you can use feature flags to **hide** the feature in one or all of the deployment rings. For example, you could enable all features in the canaries ring, and fine-tune a subset for the early adopters and production users, as shown. See [Feature Flags or Rings](#) for more information.



[LaunchDarkly](#) provides an extension for Azure DevOps Services & Team Foundation Server. It integrates with Azure Pipelines and gives you "run-time" control of features deployed with your ring deployment process.

Conclusion

Now that you've covered the concepts of rings, you should be confident to explore ways to improve your CI/CD pipelines. While the use of rings adds a level of complexity, having a game plan to address feature management and rapid customer feedback is invaluable.

Q & A

How do you know that a change can be deployed to the next ring?

Your goal should be to have a consistent checklist for the users approving a release. See aka.ms/vsarDoD for an example definition of done checklist.

How long do you wait before you push a change to the next ring?

There is no fixed duration or "cool off" period. It depends on how long it takes for you to complete all release validations successfully.

How do you manage a hotfix?

The ring deployment model allows you to process a hotfix like any other change. The sooner an issue is caught, the sooner a hotfix can be deployed, with no impact to downstream rings.

How do you deal with variables that span (shared) release environments?

Refer to [Default and custom release variables](#).

How can you manage secrets used by the pipeline?

Refer to [Azure Key Vault](#) to safeguard cryptographic keys and other secrets used by your pipelines.

Reference information

- [CI/CD pipeline examples](#)
- [Configuring your release pipelines for safe deployments](#)
- [DevOps @ Microsoft](#)

Authors: Josh Garverick, Willy Schaub | Find the origin of this article and connect with the ALM | DevOps
Rangers [here](#)

(c) 2017 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Default permissions and access for Azure DevOps

9/27/2019 • 18 minutes to read • [Edit Online](#)

[Azure DevOps Services](#) | [Azure DevOps Server 2019](#) | [TFS 2018](#) | [TFS 2017](#) | [TFS 2015](#) | [TFS 2013](#)

To use Azure DevOps features, users must be added to a security group with the appropriate permissions and granted access to the web portal. Limitations to select features are based on the *access level* and *security group* to which a user is assigned. The **Basic** access level and higher supports full access to all Azure Boards features.

Stakeholder access level provides partial support to select features, allowing users to view and modify work items, but not use all features. **Stakeholder** access is available to support free access to a limited set of features by an unlimited set of stakeholders.

The most common built-in security groups—**Readers**, **Contributors**, and **Project Administrators**—and team administrator role grant permissions to specific features.

In general, use the following guidance when assign users to an access level and security group:

- Grant **Basic** access or higher and add to the **Contributors** security group full-time workers who contribute to the code base or manage projects.
- Grant **Stakeholder** access and add to the **Contributors** security group managers or users who don't actively contribute to the code base but want to check project status and provide direction, feedback, feature ideas, and business alignment to a team. Also,
- Grant **Stakeholder** access and add to the **Project Administrators** security group users tasked with managing project resources. If they also need to contribute to the code base, then you must assign them **Basic** or higher-level access.
- Grant **Stakeholder** access and add to the **Project Collection Administrators** security group users tasked with managing organization or collection resources. If they also need to contribute to the code base, then you must assign them **Basic** or higher-level access.

To learn more about administrative tasks see [About user, team, project, and organization-level settings](#). For a complete reference of all built-in groups and permissions, see [Permissions and groups](#). For information about access levels, see [About access levels](#).

In the tables provided in this article, a indicates that the corresponding access level or security group has access to a feature by default.

For a comparison chart of Stakeholder versus Basic access, see the [Feature matrix](#). To assign or change an access level, see [Add users and assign licenses](#). If you need to [grant specific users select permissions](#), you can do so.

Dashboards, charts, reports, and widgets

You can define and manage dashboards from the web portal, **Dashboard**. For an overview of dashboard and chart features, see [Dashboards](#). You set [dashboard permissions at the team level](#) from the team dashboard page.

Users granted Stakeholder access to private projects can't view or create query charts. Stakeholder access to public projects can view and create query charts.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View work item query charts (from the Queries page)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

View dashboards (including work item query charts added to the dashboard)	<input type="checkbox"/>				
Create work item query and test tracking charts ¹			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add and configure dashboards ¹			With permissions set	<input type="checkbox"/>	<input type="checkbox"/>

Notes:

1. Public project Stakeholders have full access to all features.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View charts and dashboards	<input type="checkbox"/>				
Create work item and test tracking charts			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add and configure dashboards			With permissions set	<input type="checkbox"/>	<input type="checkbox"/>

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View team dashboard home page	<input type="checkbox"/>				
Create work item and test tracking charts			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Dashboards and charts

You can pin charts to a team dashboard **Home** page.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	PROJECT ADMINS
View work item query charts (from the Queries page)		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View dashboards (including work item query charts added to the dashboard)	<input type="checkbox"/>				
Create work item query and test tracking charts ¹			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add and configure dashboards ¹			With permissions set	<input type="checkbox"/>	<input type="checkbox"/>

Notes:

1. Public project Stakeholders have full access to all features.

Task	Stakeholders	Readers	Contributors	Team Admins	Project Admins
View charts and dashboards	<input type="checkbox"/>				
Create work item and test tracking charts			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add and configure dashboards			With permissions set	<input type="checkbox"/>	<input type="checkbox"/>

Task	Stakeholders	Readers	Contributors	Team Admins	Project Admins
View team dashboard home page	<input type="checkbox"/>				
Create work item and test tracking charts			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Power BI Integration and Analytics views

From the web portal **Analytics views**, you can create and manage Analytics views. An Analytics view provides a simplified way to specify the filter criteria for a Power BI report based on the Analytics Service data store. The Analytics Service is the reporting platform for Azure DevOps. To learn more, see [What is the Analytics Service?](#).

You set [permissions](#) for the service at the project level, and for shared Analytics views at the object level. Users with **Stakeholder** access have no access to view or edit Analytics views.

Task	Readers	Contributors	Project Admins
View Analytics	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View a shared Analytics view		<input type="checkbox"/>	<input type="checkbox"/>
Edit and delete Analytics views			<input type="checkbox"/>

Azure Boards

You can plan and track work from the web portal **Boards** hub, and using Eclipse, Visual Studio, Excel, Project, and other clients. For an overview of work tracking features, see [About Agile tools](#).

Users granted Stakeholder access are granted different access to features depending on whether it is a private or a public project. For private projects, Stakeholders have limited access to select work tracking functions, whereas for public projects, Stakeholders enjoy full access to work tracking features. To learn more, see [About access levels, Stakeholder access](#).

Work tracking

You can plan and track work from the web portal **Work** hub, and using Eclipse, Visual Studio, Excel, Project, and other clients. For an overview of work tracking features, see [About Agile tools](#).

NOTE

Team administrators can configure settings for their team's tools. Organization owners and members of the Project Administrators group can configure settings for all teams. To be added as an administrator, see [Add team administrators](#) or [Add administrators, set permissions at the project-level or project collection-level](#).

General work item feature access

You can use work items to track anything you need to track. To learn more, see [Understand how work items are used to track issues, tasks, and epics](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items, add tags to work items <i>(Stakeholders can assign existing tags to work items, but can't add new tags)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Change work item type	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Move work item to another project			<input type="checkbox"/>	<input type="checkbox"/>
Email work items	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Apply a work item template	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Delete work items (able to restore from the Recycle bin)			<input type="checkbox"/>	<input type="checkbox"/>
Permanently delete work items				<input type="checkbox"/>
Provide feedback (through the Microsoft Feedback client)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request feedback			<input type="checkbox"/>	<input type="checkbox"/>

NOTE

You can change the work item type or move work items to another project within a project collection. These features require that the data warehouse is disabled. With the data warehouse disabled, you can use the [Analytics Service](#) to support your reporting needs. To learn more about disabling the data warehouse, see [Disable the data warehouse and cube](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items, add tags to work items <i>(Stakeholders can assign existing tags to work items, but can't add new tags)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Email work items	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>

Apply a work item template	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Delete work items (able to restore from the Recycle bin)			<input type="checkbox"/>	<input type="checkbox"/>
Permanently delete work items				<input type="checkbox"/>
Provide feedback (through the Microsoft Feedback client)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request feedback			<input type="checkbox"/>	<input type="checkbox"/>

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items, add tags to work items <i>(Stakeholders can assign existing tags to work items, but can't add new tags)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Email work items	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Delete work items (able to restore from the Recycle bin)			<input type="checkbox"/>	<input type="checkbox"/>
Permanently delete work items				<input type="checkbox"/>
Provide feedback (through the Microsoft Feedback client)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request feedback			<input type="checkbox"/>	<input type="checkbox"/>

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View/open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items, add tags to work items <i>(Stakeholders can assign existing tags to work items, but can't add new tags)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Email work items	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Permanently delete work items				<input type="checkbox"/>
Provide feedback (through the Microsoft Feedback client)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request feedback			<input type="checkbox"/>	<input type="checkbox"/>

Boards feature access

You use **Boards** to implement Kanban methods. Boards present work items as cards and support quick status updates through drag-and-drop.

Task	Stakeholders	Readers	Contributors	Team Admins
View boards and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a board; update status, reorder, or reparent child tasks through drag-and-drop; update a field on a card	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add child tasks to a checklist	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign to a sprint (from card menu)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customize a board, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Task	Stakeholders	Readers	Contributors	Team Admins
View boards and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a board; update status through drag-and-drop	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Assign to a sprint	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Customize a board, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Backlogs features access

Backlogs display work items as lists. A product backlog represents your project plan and a repository of all the information you need to track and share with your team. Portfolio backlogs allow you to group and organize your backlog into a hierarchy.

Task	Stakeholders	Readers	Contributors	Team Admins
View backlogs and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a backlog <i>(Stakeholders can only add items to the bottom of the backlog)</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Use bulk edit features	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane; Assign items to a sprint using the Planning pane	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Customize a backlog, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---	--------------------------	--------------------------	--------------------------	--------------------------

Task	Stakeholders	Readers	Contributors	Team Admins
View backlogs and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a backlog <i>(Stakeholders can only add items to the bottom of the backlog)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Use bulk edit features	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Add child items to a backlog item; prioritize or reorder a backlog; parent items using the Mapping pane			<input type="checkbox"/>	<input type="checkbox"/>
Customize a backlog, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>			<input type="checkbox"/>

Sprints feature access

You use sprint tools to implement Scrum methods. The **Sprints** set of tools provide filtered views of work items that a team has assigned to specific iteration paths or sprints.

Task	Stakeholders	Readers	Contributors	Team Admins
View sprint backlogs, taskboards, and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a sprint backlog <i>(Stakeholders can add backlog items to the bottom of a sprint backlog)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a taskboard <i>(Stakeholders can add backlog items but not tasks)</i>			<input type="checkbox"/>	<input type="checkbox"/>
Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to a sprint using the Planning pane			<input type="checkbox"/>	<input type="checkbox"/>
View team capacity (work details)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Set team capacity			<input type="checkbox"/>	<input type="checkbox"/>
Use bulk edit features	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Define sprints, set sprint dates				<input type="checkbox"/>

Customize a sprint backlog or taskboard, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View sprint backlogs, taskboards, and open work items	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a sprint backlog <i>(Stakeholders can add backlog items to the bottom of a sprint backlog)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Add work items to a taskboard <i>(Stakeholders can add backlog items but not tasks)</i>			<input type="checkbox"/>	<input type="checkbox"/>
Prioritize/reorder a sprint backlog or taskboard; add child items to a backlog item; reassign items to another using drag-and-drop			<input type="checkbox"/>	<input type="checkbox"/>
View team capacity (work details)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Set team capacity			<input type="checkbox"/>	<input type="checkbox"/>
Use bulk edit features	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Define sprints, set sprint dates				<input type="checkbox"/>
Customize a sprint backlog or taskboard, configure team settings <i>(Stakeholders assigned as a team administrator or Project Administrator can configure team settings)</i>	<input type="checkbox"/>			<input type="checkbox"/>

Queries and semantic search

Queries are filtered lists of work items based on criteria that you define by using a query editor. [Adhoc searches](#) are powered by a semantic search engine.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	PROJECT ADMINS
View and run managed queries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create and save managed My queries	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Create and save managed Shared queries <i>(Stakeholders can't save Shared queries even if granted permissions)</i>				<input type="checkbox"/>
View query charts		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create query charts			<input type="checkbox"/>	<input type="checkbox"/>

Powerful semantic work-tracking search	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--	--------------------------	--------------------------	--------------------------	--------------------------

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS
View and run managed queries	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create and save managed queries <i>(Stakeholders can't save shared queries)</i>	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
View query charts		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create query charts			<input type="checkbox"/>	<input type="checkbox"/>

Delivery plans feature access

[Delivery plans](#) display work items as cards against a calendar view. This format can be an effective communication tool with managers, partners, and stakeholders for a team. Users granted **Stakeholder** access for private projects have no access to delivery plans, while users granted **Stakeholder** access for public projects has the same access as regular Contributors granted **Basic** access.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	PROJECT ADMINS
View delivery plans		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Create, edit, or delete a delivery plan <i>(Contributors can only edit or delete plans that they create)</i>			<input type="checkbox"/>	<input type="checkbox"/>
Manage permissions for a delivery plan <i>(Contributors can only manage permissions for plans that they create)</i>				<input type="checkbox"/>

Additional permissions

In addition to the permissions set at the [project level via the built-in groups](#), you can set permissions for the following objects: [area and iteration paths](#) and individual [queries and query folders](#).

Azure Repos

You can manage your source code from the web portal **Repos** hub, or using Xcode, Eclipse, IntelliJ, Android Studio, Visual Studio, or Visual Studio Code.

Stakeholders for private projects have no access to **Repos**. Stakeholders for public projects have the same access to **Repos** as **Contributors**.

Code: Source control

You can connect to your code from the web portal **Code** hub, or using Xcode, Eclipse, IntelliJ, Android Studio, Visual Studio, or Visual Studio Code. Stakeholders for private projects have no access to **Code**.

Git

You can use [Git repositories](#) to host and collaborate on your source code. For an overview of code features and functions.

Set permissions across all Git repositories by making changes to the top-level **Git repositories** entry. Individual

repositories inherit permissions from the top-level **Git Repositories** entry. Branches inherit a subset of permissions from assignments made at the repository level. For branch permissions and policies, see [Set branch permissions](#) and [Improve code quality with branch policies](#).

Task	Readers	Contributors	Build Admins	Project Admins
Clone, fetch, and explore the contents of a repository; also, can create, comment on, vote, and contribute to pull requests	✓	✓	✓	✓
Contribute to a repository, create branches, create tags, manage notes		✓	✓	✓
Create, delete, and rename repositories				✓
Edit policies, Manage permissions, Remove others' locks				✓
Bypass policies when completing pull requests, Bypass policies when pushing, Force push (rewrite history, delete branches and tags) (<i>not set for any security group</i>)				

Set permissions across all Git repositories by making changes to the top-level **Git repositories** entry. Individual repositories inherit permissions from the top-level **Git Repositories** entry. Branches inherit a subset of permissions from assignments made at the repository level. For branch permissions and policies, see [Set branch permissions](#) and [Improve code quality with branch policies](#).

By default, the project-level Readers groups have read-only permissions.

Task	Contributors	Build Admins	Project Admins
Branch Creation: At the repository level, can push their changes to branches in the repository. Does not override restrictions in place from branch policies . At the branch level, can push their changes to the branch and lock the branch.	✓	✓	✓
Contribute: At the repository level, can push their changes to branches in the repository. Does not override restrictions in place from branch policies . At the branch level, can push their changes to the branch and lock the branch.	✓	✓	✓
Note Management: Can push and edit Git notes to the repository. They can also remove notes from items if they have the Force permission.	✓	✓	✓
Tag Creation: Can push tags to the repository, and can also edit or remove tags if they have the Force permission.	✓	✓	✓
Administer: Delete and rename repositories If assigned to the top-level Git repositories entry, can add additional repositories. At the branch level, users can set permissions for the branch and unlock the branch. The Administer permission set on an individual Git repository does not grant the ability to rename or delete the repository. These tasks require Administer permissions at the top-level Git repositories entry.			✓

Rewrite and destroy history (force push): Can force an update to a branch and delete a branch. A force update can overwrite commits added from any user. Users with this permission can modify the commit history of a branch.				✓
---	--	--	--	---

The Project Collection Build Service can read from all repositories by default. Any pipeline which runs with project collection scope can potentially read any repository in the organization/collection. You can remove this permission for a repository: set "Read" to "Deny" for the Project Collection Build Service.

TFVC

[Team Foundation Version Control \(TFVC\)](#) provides a centralized version control system to manage your source control.

TASK	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS
Contribute to a centralized version control, including Code Review (Check in, label, lock, merge, pend a change)	Read only	✓	✓	✓
Check in, revise, undo, or unlock other users' changes				✓
Manage branches, manage permissions				✓

Azure Pipelines

You can define and manage your builds and releases from the web portal **Pipelines** hub. For an overview of pipelines features and functions, see [Continuous integration on any platform](#).

Build and Release

You can define and manage your builds and releases from the web portal, **Build and Release**. For an overview of pipelines features and functions, see [Continuous integration on any platform](#).

From the web portal, you can set permissions for all or individual build pipelines, release pipelines, task groups, or variable groups. See [Set build and release permissions](#).

NOTE

When the **Free access to Pipelines for Stakeholders** preview feature is enabled for the organization, Stakeholders get access to all **Build and Release** features. This is indicated by the  preview icon shown in the following table. Without this feature enabled, stakeholders can only view and approve releases. To learn more, see [Provide Stakeholders access to edit build and release pipelines](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS	RELEASE ADMINS
View release pipelines	<input type="checkbox"/>					
Define builds with continuous integration	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Define releases and manage deployments	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Approve releases	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Azure Artifacts (5 users free)	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Queue builds, edit build quality	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Manage build queues and build qualities	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	
Manage build retention policies, delete and destroy builds	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Administer build permissions	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	
Manage release permissions	<input type="checkbox"/>				<input type="checkbox"/>	<input type="checkbox"/>
Create and edit task groups	<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manage task group permissions	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Can view library items such as variable groups	<input type="checkbox"/>					
Use and manage library items such as variable groups	<input type="checkbox"/>			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	BUILD ADMINS	PROJECT ADMINS	RELEASE ADMINS
------	--------------	---------	--------------	--------------	----------------	----------------

View build and release pipelines	<input type="checkbox"/>					
Define builds with continuous integration			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Define releases and manage deployments			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Approve releases	<input type="checkbox"/>		<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Azure Artifacts (5 users free)			<input type="checkbox"/>		<input type="checkbox"/>	<input type="checkbox"/>
Queue builds, edit build quality			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Manage build queues and build qualities				<input type="checkbox"/>	<input type="checkbox"/>	
Manage build retention policies, delete and destroy builds			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Administer build permissions				<input type="checkbox"/>	<input type="checkbox"/>	
Manage release permissions					<input type="checkbox"/>	<input type="checkbox"/>
Create and edit task groups			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Manage task group permissions				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Can view library items such as variable groups		<input type="checkbox"/>				

Use and manage library items such as variable groups						
--	--	--	--	--	--	--

Azure Test Plans

Test

You can define and manage manual tests from the web portal, **Test Plans** or **Test**. For an overview of manual test features and functions, see [Testing overview](#). You set [test permissions at the project level](#) from **Project Settings>Security**.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	PROJECT ADMINS
Provide feedback using the Test & Feedback extension	✓	✓	✓	✓
Exploratory testing, view test runs		✓	✓	✓
Manage test plans and test suites Manage test configurations and test environments			✓	✓
Exploratory testing, create and delete test runs			✓	✓
Request feedback using the Test & Feedback extension			✓	✓
Azure Test Plans (formerly Test Manager, purchased separately)			✓	✓

Azure Artifacts

You can manage feeds from the web portal, **Artifacts** or **Build and release > Packages**. Feeds have three permission levels: Owners, Contributors, and Readers. Owners can add any type of identity—individuals, teams, and groups—to any permission level. To set permissions, see [Secure feeds using permissions](#).

Users granted Stakeholder or Basic access, or higher can access Azure Artifacts features.

Users granted Basic access or higher can access Azure Artifacts features. Users granted Stakeholder access have no access to Azure Artifacts.

Package management

You can manage feeds from the web portal, **Build and release > Packages**. Feeds have three levels of access: Owners, Contributors, and Readers. Owners can add any type of identity—individuals, teams, and groups—to any access level. To set permissions, see [Secure feeds using permissions](#).

Users granted Basic access or higher can access Package management features. Users granted Stakeholder access

have no access.

PERMISSION	READER	CONTRIBUTOR	OWNER
List and restore/install packages	✓	✓	✓
Push packages		✓	✓
Unlist/deprecate packages		✓	✓
Delete/unpublish package			✓
Edit feed permissions			✓
Rename and delete feed			✓

Notifications, alerts, and team collaboration tools

To manage notifications, see [Manage personal notifications](#) and [Manage team notifications](#).

NOTE

There are no UI permissions associated with managing notifications. Instead, you can manage them using the [TFS Security command line tool](#).

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ORGANIZATION OWNER/PROJECT ADMINS
Set personal notifications or alerts	✓		✓	✓	✓
Set team notifications or alerts				✓	✓
Set project-level notifications or alerts					✓
READMEs	See Note 1	✓	✓	✓	✓
View Project Wikis	✓	✓	✓	✓	✓
View Code Wikis		✓	✓	✓	✓
Provision or create a Wiki					✓
Publish Code as Wiki			✓	See Note 2	See Note 2
View the project page	✓	✓	✓	✓	✓
Edit the project page					✓
Navigate using the Project pages	✓	✓	✓	✓	✓

Request feedback		✓	✓	✓	✓
Provide feedback	✓	✓	✓	✓	✓
Powerful semantic code search	✓	✓	✓	✓	✓
Powerful semantic work tracking search	✓	✓	✓	✓	✓

Notes

1. Can view project READMEs, but not READMEs defined for a repository.
2. Project Admins or Team Admins with contribute permission can publish code as wiki. Project Admins have this permission by default.

TASK	STAKEHOLDERS	READERS	CONTRIBUTORS	TEAM ADMINS	ORGANIZATION OWNER/PROJECT ADMINS
Set personal notifications or alerts	✓		✓	✓	✓
Set team notifications or alerts				✓	✓
Set project-level notifications or alerts					✓
READMEs	See Note 1	✓	✓	✓	✓
View Project Wikis	✓	✓	✓	✓	✓
View Code Wikis		✓	✓	✓	✓
Provision or create a Wiki					✓
Publish Code as Wiki			✓	See Note 2	See Note 2
View the project page	✓	✓	✓	✓	✓
Edit the project page					✓
Navigate using the Project pages	✓	✓	✓	✓	✓
Request feedback		✓	✓	✓	✓
Provide feedback	✓	✓	✓	✓	✓
Powerful semantic code search	✓	✓	✓	✓	✓
Powerful semantic work tracking search	✓	✓	✓	✓	✓

Notes

1. Can view project READMEs, but not READMEs defined for a repository.
2. Project Admins or Team Admins with contribute permission can publish code as wiki. Project Admins have this permission by default.

Task	Stakeholders	Readers	Contributors	Team Admins	Organization Owner/ Project Admins
Set personal notifications or alerts	✓		✓	✓	✓
Set team notifications or alerts				✓	✓
Set project-level notifications or alerts					✓
Participate in Team (chat) rooms			✓	✓	✓
READMEs <i>Can view project READMEs, but not READMEs defined for a repository.</i>	Partial access	✓	✓	✓	✓
Request feedback		✓	✓	✓	✓
Provide feedback	✓	✓	✓	✓	✓

Task	Stakeholders	Readers	Contributors	Team Admins	Organization Owner/ Project Admins
Set personal notifications or alerts	✓		✓	✓	✓
Set team notifications or alerts				✓	✓
Set project-level notifications or alerts					✓
Participate in Team (chat) rooms			✓	✓	✓
Request feedback		✓	✓	✓	✓
Provide feedback	✓	✓	✓	✓	✓

Related notes

- [Add users to a project or team](#)
- [Permissions and groups reference](#)
- [About access levels](#)
- [Web portal navigation](#)

About access levels

9/19/2019 • 19 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

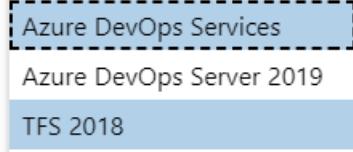
Access levels grant or restrict access to select web portal features. This is in addition to permissions granted through security groups, which provide or restrict specific tasks. Access levels enable administrators to provide their user base access to the features they need and only pay for those features.

IMPORTANT

To view the content available for your platform, make sure that you select the correct version of this article from the version selector which is located above the table of contents.

Azure DevOps Services ▾

Version



When you add a user or group to a team or project, they're automatically granted access to those features supported by the default access level and those supported by the security group to which they are added. Most users can access most features by being assigned to the **Basic** access level and **Contributors** security group. For a simplified overview of the permissions assigned to the most common groups **Readers**, **Contributors**, and **Project Administrators** as well as the **Stakeholder** access group, see [Permissions and access](#).

To add user accounts or groups to specific access levels, see [Manage users and access](#). Make sure to set each user's access level based on what you've purchased for that user.

To add user accounts or groups to specific access levels, see [Change access levels](#). Make sure to set each user's access level based on what you've purchased for that user.

Supported access levels

Assign users or groups of users to one of the following access levels:

- **Stakeholder:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic:** Provides access to most features. Assign to users with a Visual Studio Professional or MSDN Platforms subscription, and to users for whom you are paying for Basic + Test Plans access in an organization.
- **Basic + Test Plans:** Provides access to all features included in Basic, as well as Azure Test Plans. Assign to users with an Azure DevOps Server CAL or Visual Studio Professional subscription, and to users for whom you're paying for Basic access in an organization.
- **Visual Studio subscription:** Assign to users who already have a Visual Studio subscription. The system automatically recognizes the user's subscription—Visual Studio Enterprise, Visual Studio Professional, Visual Studio Test Professional, or MSDN Platform—and enables any other features that are included in their subscription level. If you assign Basic or Stakeholder, they also receive their Visual Studio subscription benefits upon sign-in.

The following table indicates those features available for each supported access level. Visual Studio Test Professional and MSDN Platform subscriptions grant access to the same features as Visual Studio Enterprise.

FEATURE	STAKEHOLDER	BASIC & VISUAL STUDIO PROFESSIONAL	BASIC + TEST PLANS & VISUAL STUDIO ENTERPRISE
Administer organization Can configure resources when also added to a security group or role: team administrator, Project Administrator, or Project Collection Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced backlog and sprint planning tools Includes full access to all backlog and sprint planning tools	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced home page Includes access to projects, work items, and pull requests defined across projects you work in	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced portfolio management Includes full access to define features and epics from a portfolio backlog or Kanban board	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile boards Includes limited access to Kanban boards . Stakeholders can't add work items, can't drag-and-drop work items to update status, and can't update fields displayed on cards.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile Portfolio Management Includes limited access to portfolio backlogs and Kanban boards . Stakeholders can't change the backlog priority order, and can't assign items to an iteration, use the mapping pane, or exercise forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analyze test results and manage machine groups Includes tracking test status and testing different configurations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Artifacts Includes full access to all Azure Artifacts features (previously referred to as package management), up to 2GB free	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Author Release Pipelines and Manage Releases Includes defining release pipelines and multi-stage continuous deployment (CD) pipelines , and using approvals and gates to control deployments ; when the Free access to Pipelines Preview feature is enabled , Stakeholders gain access to all Azure Pipelines features.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Basic backlog and sprint planning tools Includes limited access to add and modify items on backlogs and sprint backlogs and taskboards . Stakeholders can't assign items to an iteration, use the mapping pane, or forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Build Includes full access to all features to manage continuous integration and continuous delivery of software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Chart Authoring Can create work tracking query charts	<input type="checkbox"/>	<input type="checkbox"/>	
Chart Viewing Can only view work tracking query charts. Stakeholders can't view query charts from the Queries page, however can view them when added to a dashboard.	<input type="checkbox"/>	<input type="checkbox"/>	
Code Includes full access to all features to manage code using Git repositories or using Team Foundation Version Control (TFVC)	<input type="checkbox"/>	<input type="checkbox"/>	
Delivery Plans Includes full access to add and view Delivery plans.	<input type="checkbox"/>	<input type="checkbox"/>	
Request and Manage Feedback Includes full access to request and manage feedback on working software.	<input type="checkbox"/>	<input type="checkbox"/>	
Standard Features Includes working across projects , View dashboards , View wikis , Manage personal notifications . Stakeholders can't view markdown README files defined for repositories and can only read wiki pages.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test services in build and release Includes running unit tests with your builds , reviewing and analyzing test results	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test summary access to Stakeholder license Includes performing user acceptance testing and requesting Stakeholder feedback using the Test & Feedback extension	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View My Work Items Access to add and modify work items , follow work items , view and create queries , and submit, view, and change feedback responses . Stakeholders can only assign existing tags to work items (can't add new tags) and can only save queries under My Queries (can't save under Shared Queries).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View Releases and Manage Approvals Includes viewing releases and approving releases ; when the Free access to Pipelines Preview feature is enabled , Stakeholders gain access to all Azure Pipelines features.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web-based Test Case Management Includes adding test plans and test suites , creating manual test cases , and deleting test artifacts			<input type="checkbox"/>
Web-based Test Execution Includes running manual and automated tests	<input type="checkbox"/>	<input type="checkbox"/>	

- **Stakeholder:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic:** Provides access to most features. Assign to users with an Azure DevOps Server CAL, with a Visual Studio Professional subscription, and to users for whom you're paying for Basic access in an organization.
- **Basic + Test Plans:** Provides access for users who have a monthly Test Manager subscription, Visual Studio Test Professional, or MSDN Platforms subscription.

- **VS Enterprise:** Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise.

The following table indicates those features available for each supported access level.

FEATURE	STAKEHOLDER	BASIC	BASIC + TEST PLANS & VS ENTERPRISE
Administer organization Can configure resources when also added to a security group or role: team administrator, Project Administrator, or Project Collection Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced backlog and sprint planning tools Includes full access to all backlog and sprint planning tools	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced home page Includes access to projects, work items, and pull requests defined across projects you work in	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced portfolio management Includes full access to defining features and epics from a portfolio backlog or Kanban board	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile boards Includes limited access to Kanban boards . Stakeholders can't add work items, can't drag-and-drop work items to update status, and can't update fields displayed on cards.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile Portfolio Management Includes limited access to portfolio backlogs and Kanban boards . Stakeholders can't change the backlog priority order, and can't assign items to an iteration, use the mapping pane, or exercise forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analyze test results and manage machine groups Includes tracking test status and testing different configurations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Artifacts Includes full access to all Azure Artifacts features (also referred to as package management)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Author Release Pipelines and Manage Releases Includes defining release pipelines and multi-stage continuous deployment (CD) pipelines , and using approvals and gates to control deployments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Basic backlog and sprint planning tools Includes limited access to add and modify items on backlogs and sprint backlogs and taskboards . Stakeholders can't assign items to an iteration, use the mapping pane, or forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Build Includes full access to all features to manage continuous integration and continuous delivery of software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chart Authoring Can create work tracking query charts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Chart Viewing Can only view work tracking query charts; Stakeholders can't view query charts from the Queries page, however can view them when added to a dashboard.			
Code Includes full access to all features to manage code using Git repositories or using Team Foundation Version Control (TFVC)			
Delivery Plans Includes full access to add and view Delivery plans.			
Request and Manage Feedback Includes full access to request and manage feedback on working software.			
Standard Features Includes working across projects , View dashboards , View wikis , Manage personal notifications ; Stakeholders can't view markdown README files defined for repositories and can only read wiki pages.			
Test services in build and release Includes running unit tests with your builds , reviewing and analyzing test results			
Test summary access to Stakeholder license Includes performing user acceptance testing and requesting Stakeholder feedback using the Test & Feedback extension			
View My Work Items Includes limited access to add and modify work items , follow work items , view and create queries , and submit, view, and change feedback responses . Stakeholders can only assign existing tags to work items (can't add new tags) and can only save queries under My Queries (can't save under Shared Queries).			
View Releases and Manage Approvals Includes viewing releases and approving releases			
Web-based Test Case Management Includes adding test plans and test suites , creating manual test cases , and deleting test artifacts			
Web-based Test Execution Includes running manual and automated tests			
Microsoft published Azure DevOps Extensions			

- **Stakeholder:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic:** Provides access to most features. Assign to users with a CAL or with a Visual Studio Professional subscription.
- **Advanced** (legacy access level, deprecated in Azure DevOps Server 2019): Provides access to premium features. Only assign to users with a subscription to MSDN Platforms or Visual Studio Test Professional.
- **VS Enterprise:** Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise.

- **Stakeholder**: Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic**: Provides access to most features. Assign to users with a CAL or with a Visual Studio subscription.
- **Advanced** (TFS 2017): Provides access to premium features. Only assign to users with a subscription to MSDN Platforms or Visual Studio Test Professional.
- **VS Enterprise** (TFS 2017.1 and later versions): Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise.

The following table indicates those features available for each supported access level.

FEATURE	STAKEHOLDER	BASIC	ADVANCED & VS ENTERPRISE
Administer organization Can configure resources when also added to a security group or role: team administrator, Project Administrator, or Project Collection Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced backlog and sprint planning tools Includes full access to all backlog and sprint planning tools	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced home page Includes access to projects , work items, and pull requests defined across projects you work in	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced portfolio management Includes full access to defining features and epics from a portfolio backlog or Kanban board	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile boards Includes limited access to Kanban boards . Stakeholders can't add work items, can't drag-and-drop work items to update status, and can't update fields displayed on cards.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile Portfolio Management Includes limited access to portfolio backlogs and Kanban boards . Stakeholders can't change the backlog priority order, and can't assign items to an iteration, use the mapping pane, or exercise forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Analyze test results and manage machine groups Includes tracking test status and testing different configurations	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Artifacts Includes full access to all Azure Artifacts features (also referred to as package management)		<input type="checkbox"/>	<input type="checkbox"/>
Author Release Pipelines and Manage Releases Includes defining release pipelines and multi-stage continuous deployment (CD) pipelines , and using approvals and gates to control deployments	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Basic backlog and sprint planning tools Includes limited access to add and modify items on backlogs and sprint backlogs and taskboards . Stakeholders can't assign items to an iteration, use the mapping pane, or forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Build Includes full access to all features to manage continuous integration and continuous delivery of software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Chart Authoring Can create work tracking query charts	<input type="checkbox"/>	<input type="checkbox"/>	
Chart Viewing Can only view work tracking query charts; Stakeholders can't view query charts from the Queries page, however can view them when added to a dashboard.	<input type="checkbox"/>	<input type="checkbox"/>	
Code Includes full access to all features to manage code using Git repositories or using Team Foundation Version Control (TFVC)	<input type="checkbox"/>	<input type="checkbox"/>	
Delivery Plans Includes full access to add and view Delivery plans.	<input type="checkbox"/>	<input type="checkbox"/>	
Request and Manage Feedback Includes full access to request and manage feedback on working software.	<input type="checkbox"/>	<input type="checkbox"/>	
Standard Features Includes working across projects , View dashboards , View wikis , Manage personal notifications . Stakeholders can't view markdown README files defined for repositories and can only read wiki pages.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Team rooms Requires TFS 2017 or earlier versions. Deprecated for TFS 2018 and later versions.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Test summary access to Stakeholder license Includes performing user acceptance testing and requesting Stakeholder feedback using the Test & Feedback extension	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View My Work Items Includes limited access to add and modify work items , follow work items , view and create queries , and submit, view, and change feedback responses . Stakeholders can only assign existing tags to work items (can't add new tags) and can only save queries under My Queries (can't save under Shared Queries).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View Releases and Manage Approvals Includes viewing releases and approving releases	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web-based Test Case Management Includes adding test plans and test suites , creating manual test cases , and deleting test artifacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web-based Test Execution Includes running manual and automated tests	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

- **Stakeholder/Limited:** Provides partial access, can be assigned to unlimited users for free. Assign to users with no license or subscriptions who need access to a limited set of features.
- **Basic/Standard:** Provides access to most features. Assign to users with a CAL or with a Visual Studio subscription.
- **Advanced/Full:** Provides access to premium features. Assign to users with a subscription to Visual Studio Enterprise, Visual Studio Test Professional or MSDN Platforms.

The following table indicates those features available for each supported access level.

FEATURE	STAKEHOLDER (LIMITED)	BASIC (STANDARD)	ADVANCED (FULL)

Administer organization Can configure resources when also added to a security group or role: team administrator, Project Administrator, or Project Collection Administrator	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced backlog and sprint planning tools Includes full access to all backlog and sprint planning features.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Advanced home page Includes access to projects , work items , and pull requests defined across projects you work in	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile boards Includes limited access to Kanban boards . Stakeholders can't add work items and can't drag-and-drop work items to update status.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Agile Portfolio Management Includes full access to portfolio backlogs and Kanban boards	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Basic backlog and sprint planning tools Includes limited access to add and modify items on backlogs and sprint backlogs and taskboards . Stakeholders can't assign items to an iteration, use the mapping pane, or forecasting.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Build Includes full access to all features to manage continuous integration and continuous delivery of software	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chart Authoring Can create work tracking query charts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Chart Viewing Can only view work tracking query charts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Code Includes full access to all features to manage code using Git repositories or using Team Foundation Version Control (TFVC)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Request and Manage Feedback Includes full access to request and manage feedback on working software.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Standard Features Includes working across projects , viewing dashboards , and managing personal notifications ; Stakeholders have no access to repositories.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Team rooms	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
View My Work Items Includes limited access to add and modify work items , follow work items , view and create queries , and submit, view, and change feedback responses . Stakeholders can only assign existing tags to work items (can't add new tags) and can only save queries under My Queries (can't save under Shared Queries).	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Web-based Test Management Includes adding and executing test plans and test suites and manual test cases , and deleting test artifacts	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Stakeholder access

With Stakeholder access, users can create and modify work items and create and save queries. They have limited access to many of the Azure Boards features. They also can view and approve release pipelines and perform administrative tasks when granted administrative permissions or added to an administrative group.

To get started as a Stakeholder, see [Get started as a Stakeholder](#).

Public versus private feature access

Stakeholder access grants access to features differently depending on whether you're working from a private or a public project. To learn more about public projects, see [What is a public project?](#)

SERVICE, APPLICATION, OR SETTING	PRIVATE PROJECT	PUBLIC PROJECT
Dashboards	Partial access	Full access
Wiki	Partial access	Full access
Azure Boards	Partial access	Full access
Azure Repos	No access	Full access
Azure Pipelines	Full access	Full access
Azure Test Plans	No access	No access
Azure Artifacts	Full access	Full access
Notifications	Full access	Full access
Semantic search	Full access	Full access
Project settings	Partial access	Partial access
Organization settings	Partial access	Partial access

Features not available to users with Stakeholder access

If a Stakeholder needs access to one or more of the following features—which support the daily work of product owners, team leads, developers, testers, and project administrators—you need to grant them **Basic** access.

For Private projects:

- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Pipelines**, **Repos** or **Test Plans**.

For Public projects:

- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Repos** or **Test Plans**.
- Change the priority of an item within a backlog
- Delete work items or move work items to another project
- Create shared queries, view charts, and modify the home page

- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Pipelines, Repos** or **Test Plans**.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features under **Code, Build and Release** or **Test**.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify the home page
- View Delivery Plans (a Marketplace extension)
- Access the full set of features provided under **Code, Build and Release** or **Test**
- Participate in team rooms, which capture interactive, detailed conversations about the project.
- Change the priority of an item within a backlog
- Delete work items
- Create shared queries, view charts, and modify the home page
- Access the full set of features provided under **Code, Build and Release** or **Test**
- Participate in team rooms, which capture interactive, detailed conversations about the project.

NOTE

Stakeholders that choose a feature that's not available to them receive an error message indicating that they don't have permissions to complete the task.

Visual Studio subscription access

Visual Studio subscribers are entitled to **Visual Studio subscription** features as a subscriber benefit. When you add those users, be sure to assign them the **Visual Studio subscription** access level.

The system automatically recognizes their subscription and enables any other features that are included, based on their subscription level.

VS Enterprise access

Visual Studio Enterprise subscribers are entitled to **VS Enterprise** access as a subscriber benefit. When you add those users, be sure to assign them the **VS Enterprise** access level.

With VS Enterprise access, users have access to any fee-based, Marketplace extension published by Microsoft Marketplace extension published by Microsoft that is included for active Visual Studio Enterprise subscribers.

For TFS 2017.2 and later versions, assign **VS Enterprise** to those users for whom you've purchased Visual Studio Enterprise. These include a TFS CAL plus the rights to access VS Enterprise features. (For users with MSDN Platforms subscriptions or Test Professional, assign the Basic access level and the Test Manager extension for Azure Test Plans.) To learn more, see [Assign paid extension access to users](#). For example, for users with Visual Studio Test Professional or Visual Studio Enterprise, assign them [access to the Test Manager extension for Azure Test Plans](#).

Advanced access

Users assigned Advanced access can manage test cases when you have [purchased the Test Manager extension](#) for

Azure Test Plans and assigned to the user accounts to gain full access to [Web-based test case management tools](#).

Users assigned Advanced access have all the Basic features, plus [web-based test case management tools](#). You can [buy monthly access](#) or add users who already have a Visual Studio Test Professional with MSDN or MSDN Platforms subscription.

For TFS 2017 and earlier versions, you should assign the **Advanced** level to those users for whom you've purchased the full Test feature set. Here are the purchasing options:

- Higher-level Visual Studio subscriptions: Visual Studio Test Professional, Visual Studio Enterprise, or MSDN Platforms subscriptions. These include a TFS CAL plus the rights to access the full set of Test features.
- A paid Azure DevOps user (which includes a TFS CAL) plus the [Test Manager extension](#).

For TFS 2017.2, Assign **Advanced** access to those users for whom you've purchased MSDN Platforms or Visual Studio Test Professional subscriptions. These include a TFS CAL plus the rights to access Test Manager. To learn more, see [Get extensions for TFS](#), [Assign paid extension access to users](#).

NOTE

With TFS 2017.1, the Advanced access level was temporarily disabled. Updating to TFS 2017.2 re-enables it. If you are on TFS 2017.1 and have users with Visual Studio Test Professional or MSDN Platforms subscriptions, you should assign them Basic access. In addition, you need to open **Users** for the project collections in which they are a member and [assign them the Test Manager extension for Azure Test Plans](#). To learn more, see [Buy access to TFS or TFS Test](#).

What features are available to users who are added to two different access levels?

If a user belongs to a group that has **Basic** access and another group that has **VS Enterprise** access, the user has access to all features available through **VS Enterprise**, which is a superset of **Basic**.

Service account access

Azure DevOps Server [service accounts](#) are added to the default access level. If you make Stakeholder the default access level, you must add the service accounts to Basic or Advanced/VS Enterprise access.

Service accounts don't require a CAL or other purchase.

Related articles

- [Free access to Pipelines Preview](#)
- [Manage users and access](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)
- [Change access levels](#)
- [Export a list of users and their access levels](#)
- [Default permissions and access](#)
- [Compare features between plans](#)

Navigate in Visual Studio Team Explorer

8/1/2019 • 7 minutes to read • [Edit Online](#)

Visual Studio 2019 | Visual Studio 2017 | Visual Studio 2015

You use Team Explorer to coordinate your code efforts with other team members to develop a software project. In addition, you can manage work and that is assigned to you, your team, or your projects. Team Explorer is a plug-in that installs with Visual Studio and Team Explorer Everywhere is a plug-in that installs with Eclipse. Developers can effectively collaborate using Team Explorer connected to projects hosted on Azure DevOps Services or an on-premises Azure DevOps Server (previously named Team Foundation Server (TFS)).

TIP

You can install the latest version of Visual Studio clients from the [Visual Studio downloads page](#).

Additional options for connecting to Azure DevOps Services or TFS include:

- [Team Explorer Everywhere](#)
- [Azure DevOps Plugin for Android Studio](#)
- [Azure DevOps Plugin for IntelliJ](#)
- [Visual Studio Code](#)

For information about compatibility among client and server versions, see [Requirements and compatibility](#).

If you don't need Visual Studio, but want to connect to a project in Azure DevOps, you can install the free [Visual Studio Community](#).

Prerequisites

- You must have a project in Azure DevOps. If you need to add a project, see [Create a project](#).
- You must be a member of the project you connect to. To get added, see [Add users to a project or team](#).

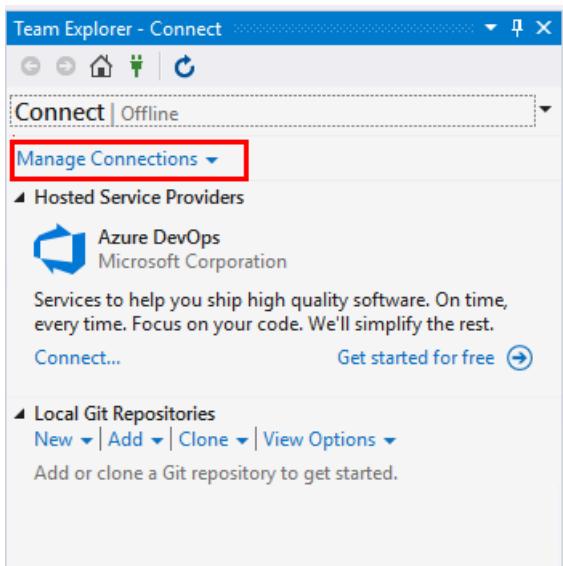
Connect to a project or repository

Team Explorer connects Visual Studio to projects in Azure DevOps. You can manage source code, work items, and builds. The operations available to you depend on which source control option—Git or Team Foundation version control (TFVC)—was selected to manage source code when the project was created.

TIP

If you open Visual Studio and the Team Explorer pane doesn't appear, choose the **View>Team Explorer** menu option from the tool bar.

From the **Connect** page, you can select the projects you want to connect to and quickly switch connection to a different project and or repository. For details, see [Connect to a project](#).



The Git and TFVC repos support different pages and functions. For a comparison of the two version control systems, see [Choosing the right version control for your project](#).

Git version control and repository

The following images show the pages available when you connect to a Git repository from Team Explorer.

VISUAL STUDIO 2019	VISUAL STUDIO 2017	VISUAL STUDIO 2015
A screenshot of the Visual Studio 2019 Team Explorer Home page. It shows a 'Fabrikam Fiber' connection under 'Azure DevOps'. Below it is a 'Project' section with 'Web Portal' and 'Task Board' links, followed by a list of git operations: Changes, Branches, Pull Requests, Sync, Tags, Work Items, Builds, and Settings.	A screenshot of the Visual Studio 2017 Team Explorer Home page. It shows a 'Fabrikam Git' connection under 'Visual Studio Team Services'. Below it is a 'Project' section with 'Web Portal' and 'Task Board' links, followed by a list of git operations: Changes, Branches, Pull Requests, Sync, Tags, Work Items, Builds, and Settings.	A screenshot of the Visual Studio 2015 Team Explorer Home page. It shows a 'Fabrikam Fiber' connection. Below it is a 'Project' section with 'Clone Repository', 'Web Portal', 'Task Board', and 'Team Room' links, followed by a list of git operations: Changes, Branches, Pull Requests, Sync, Work Items, Builds, and Settings.

To learn more about each page, see the following articles.

HOME & BUILDS	GIT VERSION CONTROL	WORK ITEMS
---------------	---------------------	------------

<p>Home</p> <ul style="list-style-type: none"> • Web portal • Task Board • Team Room <p>Builds</p> <ul style="list-style-type: none"> • Create build pipelines • View and manage builds • Manage the build queue • Install Continuous Delivery (CD) Tools for Visual Studio • Configure and execute Continuous Delivery (CD) for your app 	<ul style="list-style-type: none"> • Create a new repo • Clone an existing repo • Changes: Save work with commits • Branches: Create work in branches • Pull Requests: Review code with pull requests • Sync: Update code with fetch and pull • Tags: Work with Git tags • Git preferences • Git command reference 	<p>Default experience (Visual Studio 2019 only)</p> <ul style="list-style-type: none"> • View and add work items • Set the Work Items experience in Visual Studio <p>Legacy experience (All versions of Visual Studio)</p> <ul style="list-style-type: none"> • Add work items • Query editor • Query folders • Query permissions • Open query in Excel • Open query in Project • Email query results using Outlook • Create reports from query in Excel (TFS only)
---	--	---

Team Foundation version control

The following images show the pages available when you connect to a TFVC repository from Team Explorer.

VISUAL STUDIO 2019	VISUAL STUDIO 2017	VISUAL STUDIO 2015

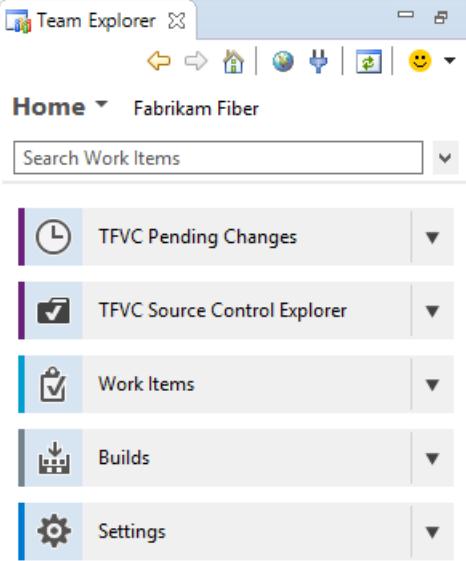
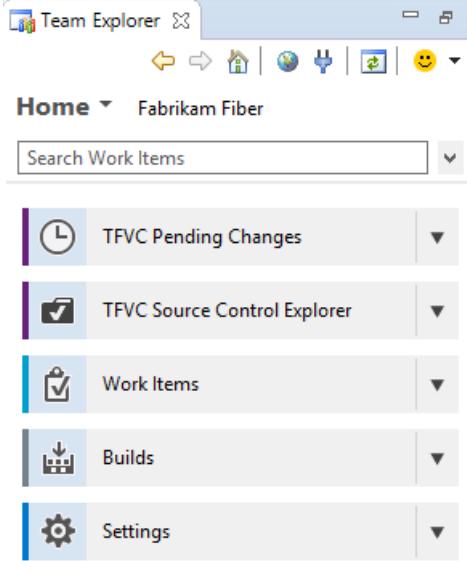
To learn more about each page, see the following articles.

HOME & BUILDS	TFVC	WORK ITEMS
-----------------------------------	----------------------	----------------------------

<p>Home</p> <ul style="list-style-type: none"> • Web portal • Task Board • Team Room <p>Builds</p> <ul style="list-style-type: none"> • Create build pipelines • View and manage builds • Manage the build queue • Install Continuous Delivery (CD) Tools for Visual Studio • Configure and execute Continuous Delivery (CD) for your app 	<ul style="list-style-type: none"> • Configure workspace • My Work: Suspend/resume work Code review • Pending Changes: Manage pending changes Find shelvesets Resolve conflicts • Source Control Explorer: Add/view files and folders • Add Check-In Policies • Version control commands 	<p>Default experience (Visual Studio 2019 only)</p> <ul style="list-style-type: none"> • View and add work items • Set the Work Items experience in Visual Studio <p>Legacy experience (All versions of Visual Studio)</p> <ul style="list-style-type: none"> • Add work items • Query editor • Query folders • Query permissions • Open query in Excel • Open query in Project • Email query results using Outlook • Create reports from query in Excel (TFS only)
---	---	---

Team Explorer plug-in for Eclipse

If you work in Eclipse or on a non-Windows platform, you can [install the Team Explorer plug-in for Eclipse](#). Once installed, you can share your Eclipse projects by adding them to Azure DevOps Services or TFS using [Git](#) or [TFVC](#).

<p>HOME PAGE WITH GIT (ECLIPSE)</p>  <p>The screenshot shows the Eclipse Team Explorer interface with the 'Home' tab selected. The main area displays a 'Search Work Items' bar and a list of project-related links: 'TFVC Pending Changes', 'TFVC Source Control Explorer', 'Work Items', 'Builds', and 'Settings'. Each link has a small icon and a dropdown arrow.</p>	<p>HOME PAGE WITH TFVC (ECLIPSE)</p>  <p>The screenshot shows the Eclipse Team Explorer interface with the 'Home' tab selected. The main area displays a 'Search Work Items' bar and a list of project-related links: 'TFVC Pending Changes', 'TFVC Source Control Explorer', 'Work Items', 'Builds', and 'Settings'. Each link has a small icon and a dropdown arrow.</p>
---	---

To learn more about each page, see the following articles.

HOME & BUILDS	VERSION CONTROL	WORK ITEMS
---------------	-----------------	------------

<p>Home</p> <ul style="list-style-type: none"> • Web portal <p>Builds</p> <ul style="list-style-type: none"> • Create build pipelines • View and manage builds • Manage the build queue • Install Continuous Delivery (CD) Tools for Visual Studio • Configure and execute Continuous Delivery (CD) for your app 	<p>Git repo</p> <ul style="list-style-type: none"> • Share your code • Git preferences • Git command reference <p>TFVC repo</p> <ul style="list-style-type: none"> • Share your code • Pending changes • Source Control Explorer • Add Check-In Policies • Version control commands 	<ul style="list-style-type: none"> • Add work items • Query editor • Query folders • Query permissions
--	---	--

Reports

NOTE

Some pages, such as **Reports**, only appear when an on-premises TFS is configured with the required resources, such as SQL Server Reporting Services and SharePoint.

The **Reports** page opens the [Reporting Services report site](#). This page appears only when your project has been configured with SQL Server Analysis Services and Reporting Services. Also, the option to **Create Report in Microsoft Excel** appears only when reporting has been configured for the project.

If your project is missing one or more pages, you may be able to [add functionality to your on premises TFS deployment](#).

Reports and Documents

NOTE

Some pages, such as **Reports** and **Documents**, only appear when an on-premises TFS is configured with the required resources, such as SQL Server Reporting Services and SharePoint.

The **Reports** page opens the [Reporting Services report site](#). This page appears only when your project has been configured with SQL Server Analysis Services and Reporting Services. Also, the option to **Create Report in Microsoft Excel** appears only when reporting has been configured for the project.

From the **Documents** page, you can [open project portal](#) and [manage documents and document libraries](#). This page appears only if your project has been configured with a SharePoint Products portal.

If your project is missing one or more pages, you may be able to [add functionality to your on premises TFS deployment](#).

Settings

From the **Settings** page, you can configure administrative features for either a project or project collection. To learn more about each page, see the following articles. Most of the links open to a web portal administration page. Not all settings are available from the Team Explorer plug-in for Eclipse.

PROJECT	PROJECT COLLECTION	OTHER
---------	--------------------	-------

- Security, Group Membership
- Security, Source Control (TFVC)
- Work Item Areas
- Work Item Iterations
- Portal Settings
- Project Alerts

- Security, Group Membership
- Source Control (TFVC)
- Process Template Manager

- Git Global Settings
- Git Repository Settings

To learn more about settings, see [About team, project, and organizational-level settings](#).

Refresh Team Explorer or Team Explorer Everywhere

If data doesn't appear as expected, the first thing to try is to refresh your client. Refreshing your client updates the local cache with changes that were made in another client or in TFS. To refresh Team Explorer, do one of the following actions:

- To refresh a page that you are currently viewing, choose  Refresh icon in the menu bar (or choose the F5 key).
- To refresh the project you currently have selected, choose  Home, and then choose  Refresh icon (or choose the F5 key).
- To refresh the set of teams defined for the project that you currently have selected, choose the Connect icon, and then choose  Refresh icon (or choose the F5 key).

To avoid potential errors, you should refresh your client application under the following circumstances:

- Process changes are made.
- Work item type definitions are added, removed, renamed, or updated.
- Area or iteration paths are added, removed, renamed, or updated.
- Users are added to or removed from security groups, or permissions are updated.
- A team member adds a new shared query or changes the name of a shared query.
- A build pipeline is added or deleted.
- A team or project is added or deleted.

Resolve images that don't display in Team Explorer

If an inline image isn't displayed in a work item form that you view in Visual Studio Team Explorer, but the image is displayed in the web portal, your credentials might have expired. You can resolve this by completing the following steps:

1. In Visual Studio, select **View > Other Windows > Web Browser** (or use the shortcut **Ctrl+Alt+R**).
2. In the web browser, locate your organization.
3. Sign in with your credentials.
4. Refresh your work item in Team Explorer.

Related articles

- [Troubleshoot connection](#)
- [Create a project](#)

Additional tools provided with TFS Power Tools

By installing [TFS Power Tools](#), you gain access to these additional tools through the Team Explorer plug-in for Visual Studio:

- Process Template Editor
- Additional check-in policies for Team Foundation Version Control

- Team Explorer enhancements including Team Members
- Team Foundation Power Tool Command Line
- Test Attachment Cleaner
- Work Item Templates

Additional requirements may apply.

FAQs about signing up and getting started

9/27/2019 • 3 minutes to read • [Edit Online](#)

Azure DevOps Services | Azure DevOps Server 2019 | TFS 2018 | TFS 2017 | TFS 2015 | TFS 2013

Signing up for Azure DevOps is now easier than ever - it's a two-minute process. See the following FAQs, which contain links for getting started.

How do I sign up for the cloud?

- [Sign up and get started in Azure DevOps Services - a two-minute process.](#)
- You can also sign up and get started with only a single service in Azure DevOps:
 - [Azure Pipelines](#)
 - [Azure Repos](#)
 - [Azure Boards](#)

How do I get started on-premises?

- Download and install [Azure DevOps Server](#)

How do I get started on-premises?

- To get started with an on-premises instance, download and install the [latest version of TFS](#).
- [Configure the installation](#), which creates a default collection.
- If you need to create a project, [create one on-premises](#).
- If you don't have access to the project, [get invited to the team](#).
- If it's your first time connecting to a project, see [Connect to a project](#).

How do I connect with a client tool?

Go to one of the following pages to download a version of Visual Studio or client tool plug-in that supports connecting to a project:

- [Visual Studio](#)
- [Eclipse/Team Explorer Everywhere](#)
- [Android Studio with the Azure DevOps Services Plugin for Android Studio](#)
- [IntelliJ with the Azure DevOps Services Plugin for IntelliJ](#)
- [Visual Studio Code](#)

How much does Azure DevOps cost?

See the following links for pricing:

- [Azure DevOps Services pricing](#)
- [Azure DevOps on-premises pricing](#)
- [Azure Pipelines only pricing](#)

How do I share code?

See about [sharing code](#).

How do I track work?

See [Plan and track work](#).

What do I do as an admin?

See [Administrator roles](#).

What client-server compatibility issue are there?

See [Requirements and compatibility](#).

Can Stakeholders who don't use Visual Studio participate on our team?

Yes. You can provide access to Stakeholders who have no client access license for the following activities:

- **Stakeholder access:** This view allows anyone on your team to check project status and provide feedback. Stakeholders can [track project priorities and provide direction, feature ideas, and business alignment to a team](#).
To grant Stakeholders access, add them to the [Stakeholder access group](#).
- **Provide feedback:** To allow your Stakeholders to provide feedback, you must [grant them specific permissions](#).

Are there other clients that connect to Azure DevOps? Are there other tools I can use?

Yes. You can connect to a project from one of the following clients:

- [Excel](#) (requires Team Foundation add-in)
- [Project](#) (requires Team Foundation add-in)
- [Project Professional](#)
- [PowerPoint Storyboarding](#) (requires Team Foundation add-in)
- [Azure Test Plans](#)
- [Test & Feedback extension \(previously called the Exploratory Testing extension\)](#)
- [Microsoft Feedback Client](#)

NOTE

Native support for integrating TFS with Project Server is deprecated for TFS 2017. However, synchronization support is provided by a third party. See [Synchronize TFS with Project Server](#) for details.

Test Manager is deprecated for TFS 2017.

You can also find several open-source clients that have been added to [Marketplace extensions](#). For example, you can install extensions to Visual Studio that support additional features:

- For TFS 2017 and later versions, you can [install the TFS Process Template editor from the Visual Studio Marketplace](#). You can use this version of the Process Editor to modify the old-style work item forms. You can't use it to edit forms associated with the [new web forms](#).
- For TFS 2015 and earlier versions, you can install [TFS Power Tools](#). TFS Power Tools provides enhancements,

tools, and command-line utilities that support increased productivity.

NOTE

Team Foundation Server Power Tools is deprecated for TFS 2017 and later versions.

Why can't I connect to a project?

See [Troubleshoot connection](#).

Related articles

- [Essential services](#)
- [Client-server tools](#)
- [Software development roles](#)
- [Azure DevOps Support](#)
- [Live chat](#) (English only)

Azure DevOps Service Status

8/1/2019 • 3 minutes to read • [Edit Online](#)

Introduction

We have a team of engineers around the world who look after the health of Azure DevOps 24 hours a day. Their primary goal is to ensure that our users are productive and successful with our service at all times. From time to time, like any online service, our service experiences performance slowdowns and stability issues. In these cases, we aim to respond quickly to restore the service, and also make it a top priority to communicate the incident status and our next steps to mitigate the issue on the [Azure DevOps Service Status Portal](#).

If you are experiencing a problem with any of our Azure DevOps services, you can check the service health to determine if we are already working on the issue before you contact our support team or spend time troubleshooting. Many of the events we post here are based on our Customer Impact Assessment (CIA). CIA is modeled on the same telemetry we use in our [availability model](#) that measures real customer representing both reliability and performance.

Concepts

Severity

The severity of a service health event is based on the number of customers affected by the issue. Typically, the highest severity events impact a large percentage of our customers and render some parts of the product unusable. These events are represented as "Unhealthy" in the service status portal. Lower severity events affect the performance of a specific feature in a service but does not make the service broadly unavailable. These events are reflected as "Degraded" in the service status portal.

Services in a Product Suite

[Azure DevOps](#) is a product suite of service offerings. For more information about pricing and acquisition, see the [pricing and acquisition page](#) for more information on how to access different services in the suite. The [geographic region](#) is indicative of where the user's account is hosted in the cloud and the data residency, sovereignty, compliance, and resilience requirements are honored within the geographical boundaries.

In addition to the list of specific Azure DevOps services, the matrix also displays two other categories: Core and Other. The Core category encompasses the set of features that are fundamental to all five services, such as authentication or the web portal. The Other category is the list of features that complement the suite, such as extensions.

Service Health Portal

Service Health Matrix

The service status portal provides a two-dimensional matrix view of active events mapped to a given service and geography. To help clarify which specific aspects of the service are affected, we communicate impact of each of these services by geographic region in the service matrix.

Event Logs

More information on active events are found in the form of an "event log" that's displayed on the top of the service health matrix. Each log will have additional associated information in the form of the impacted service and geography as well as the duration of a specific event. You can click through the provided hyperlink to navigate into the event log, which provides detailed information on the event being investigated.

The [Status history section](#) provides a view into current active events as well as past events. You can also filter the logs to adjust the scope of your search into past events. In addition, you can use the REST API build automated alerting solutions to help you stay on top of events with Azure DevOps services.

You can use [the RSS feed](#) to subscribe and receive information in your feed reader.

Using REST APIs

For users who are looking to build an automated solution to monitor the infrastructure incidents, we provide REST APIs to retrieve the current health status of each of the Azure DevOps services. Stay tuned for more information.

Get Help

If you're experiencing an issue with Azure DevOps and see a corresponding event communicated on the service health portal, be assured that we're working to restore normal operations of the service. You don't need to take any further action to notify us. However, if you don't see your issue reported on the Azure DevOps service health page, refer to the resources available on [Azure DevOps support page](#). That page has resources to report bugs, interact with a friendly support bot, and contact our support team for additional assistance.

Service limits and rate limits

8/1/2019 • 2 minutes to read • [Edit Online](#)

Learn which service limits and rate limits that all projects and organizations are subject to.

Work items

- A long text field can contain 1M characters.
- You can't assign more than 100 tags to a work item.
- You can't add more than 1,000 links to a work item.
- You can't add more than 100 attachments to a work item.
- You can't add an attachment size larger than 60 MB to a work item.
- You can have up to 1,000 tasks on a task board
- You can have up to 10,000 work items on a backlog
- You are limited to 5,000 teams in a project
- You can't create more than 150,000 tag definitions per project

Queries

- The execution time limit for queries is 30 seconds. See [optimization best practices](#) to improve query performance.
- Query results are limited to 20,000
- Queries are limited in length to 32,000 characters

Process customization

When customizing the work item types (WITs) defined in the Inheritance or Hosted XML process models, be aware of the limits placed on objects defined in this topic.

- See [Process related limits](#) for details

Wiki

Wikis defined for a project are limited to 1 GB per git repository.

TIP

To derive the size of a wiki/git repository, download the repo to your local computer, unzip the file, and then open the [Properties](#) for the corresponding folder.

Rate limiting

Azure DevOps Services, like many Software-as-a-Service solutions, uses multi-tenancy to reduce costs and to enhance scalability and performance. This leaves users vulnerable to performance issues and even outages when other users of their shared resources have spikes in their consumption. To combat these problems, Azure DevOps Services limits the resources individuals can consume and the number of requests they can make to certain commands. When these limits are exceeded, subsequent requests may be either delayed or blocked.

See [Rate limits documentation](#) for details

Data Import

- Limited to 300 projects per collection
- See [data import documentation](#) for details

Next steps

- [Review work tracking object limits](#)

Launch Visual Studio via Azure DevOps

8/1/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

When you first open [Visual Studio 2015](#), you can sign in and connect to [Azure DevOps](#).

If you've already gone through Visual Studio sign-in, or you're using Visual Studio 2017, then [learn how](#) to connect to Azure DevOps from the Team Explorer window.

After you're connected, you can store or share code in free, unlimited, private, cloud-based Git repositories or Team Foundation Version Control. Organize and manage your work by using Agile tools for DevOps, continuous integration, and continuous delivery so your team can build often, test early, and ship faster.

To set up Visual Studio without Azure DevOps, learn how to [get started](#). To host your own server, learn how to [install and set up Azure DevOps Server](#).

Azure DevOps is free for [up to five users with access to Basic features](#) and for unlimited [Visual Studio subscribers](#) and [Stakeholders who can access limited features](#). Learn [what else you get with Azure DevOps](#). If you want, you can also use Azure DevOps with any IDE or code editor, like the following:

- [Eclipse, Android Studio, or IntelliJ](#)
- Xcode (see [Git](#) or [TFVC](#))
- [Visual Studio Code](#)

How do I set up Visual Studio 2015 for Azure DevOps during sign in?

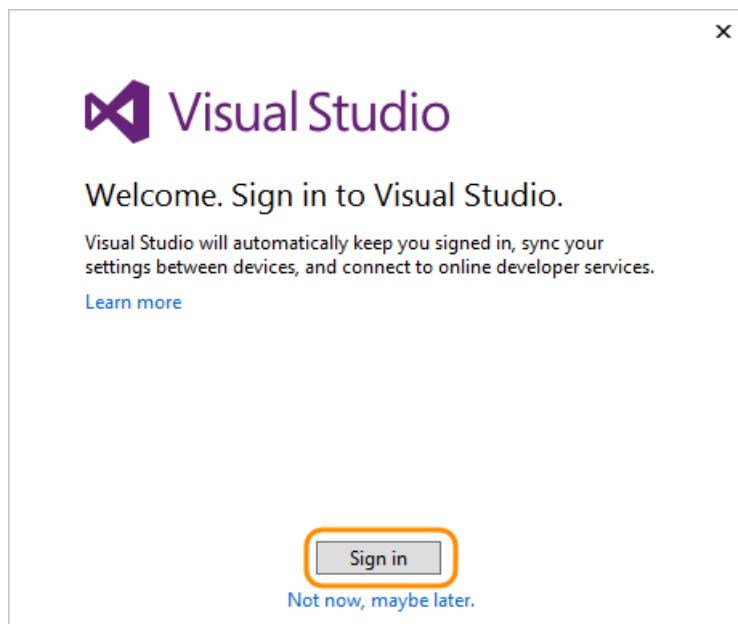
1. [Download and install Visual Studio](#), if you don't have the version you want already. [Which versions can I use with Azure DevOps?](#)

If you have a Visual Studio subscription that includes the Visual Studio IDE, get the version that's available with your subscription.

2. Start Visual Studio, and then sign in to create your profile.

This profile saves your settings and roams with you when you sign in to Visual Studio on any computer.

[Why else should I sign in?](#) If you're a Visual Studio subscriber, use the sign-in address for your subscription.



[Can't sign in?](#)

3. Enter your sign-in address, and then enter your password.
4. Add your Visual Studio profile details. You have to do this only once.

A screenshot of the Visual Studio profile details form. It includes fields for "Full name" (Jamal Hartnett), "Contact e-mail" (jamalhartnett@outlook.com), "Phone number" (empty), and "Country/Region" (United States). The "Country/Region" field has a dropdown arrow.

5. Give your organization a name, and confirm its location.

A screenshot of the Visual Studio organization creation form. It shows a "Create a Visual Studio Team Services site (optional)" field containing "https://fabrikam.visualstudio.com". Below it, a note says "Your account will be hosted in the **South Central US** region." A "Change options" link is available. A note about Microsoft's use of contact information is present, along with a statement about agreeing to the Terms of Service and Privacy Statement. A "Continue" button is at the bottom.

[How can I create an organization later or change its location?](#)

6. Create your first project to store your code, work items, backlog, builds, tests, and other assets. Name your project, select a process to organize your work, and choose the version control to manage your code.



Visual Studio

Create your first team project

Welcome. Your account, <https://fabrikam.visualstudio.com/>, is created and ready to go. Now create your first team project where you'll host your code and backlog. [Learn more](#)

Project name: *

Process template: *

Version control: *

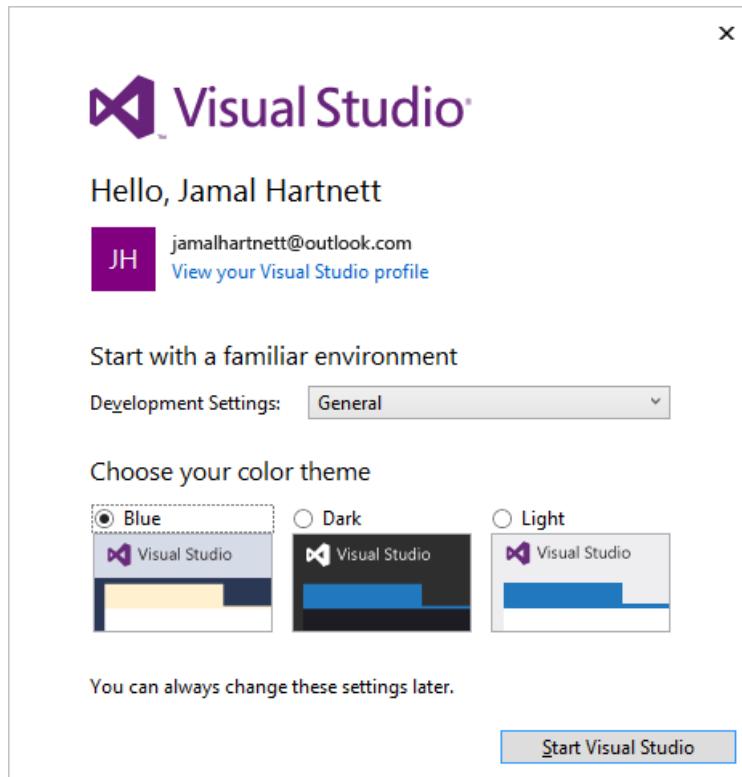
 Git [i](#)  Team Foundation Version Control [i](#)

Create a README.md file to describe this project

[Not now, maybe later](#) [Create project](#)

Not sure which to choose? Learn which [process](#) and version control ([Git](#) or [Team Foundation Version Control](#)) work best for you.

7. If you're a new Visual Studio user, you can change your settings here, or change them later in Visual Studio options.



These changes are saved with your profile, and your settings roam with you wherever you sign in.

8. To view your new organization, sign in to <https://dev.azure.com/{yourorganization}>.

[Having browser problems?](#)

Next steps

[Add users to your organization](#)

Related articles

- Add code to [Git](#) or [Team Foundation Version Control](#).
- [Create your backlog](#) to organize your work, [manage your process](#), or [customize your process](#).

Azure DevOps Service Status

8/1/2019 • 3 minutes to read • [Edit Online](#)

Introduction

We have a team of engineers around the world who look after the health of Azure DevOps 24 hours a day. Their primary goal is to ensure that our users are productive and successful with our service at all times. From time to time, like any online service, our service experiences performance slowdowns and stability issues. In these cases, we aim to respond quickly to restore the service, and also make it a top priority to communicate the incident status and our next steps to mitigate the issue on the [Azure DevOps Service Status Portal](#).

If you are experiencing a problem with any of our Azure DevOps services, you can check the service health to determine if we are already working on the issue before you contact our support team or spend time troubleshooting. Many of the events we post here are based on our Customer Impact Assessment (CIA). CIA is modeled on the same telemetry we use in our [availability model](#) that measures real customer representing both reliability and performance.

Concepts

Severity

The severity of a service health event is based on the number of customers affected by the issue. Typically, the highest severity events impact a large percentage of our customers and render some parts of the product unusable. These events are represented as "Unhealthy" in the service status portal. Lower severity events affect the performance of a specific feature in a service but does not make the service broadly unavailable. These events are reflected as "Degraded" in the service status portal.

Services in a Product Suite

[Azure DevOps](#) is a product suite of service offerings. For more information about pricing and acquisition, see the [pricing and acquisition page](#) for more information on how to access different services in the suite. The [geographic region](#) is indicative of where the user's account is hosted in the cloud and the data residency, sovereignty, compliance, and resilience requirements are honored within the geographical boundaries.

In addition to the list of specific Azure DevOps services, the matrix also displays two other categories: Core and Other. The Core category encompasses the set of features that are fundamental to all five services, such as authentication or the web portal. The Other category is the list of features that complement the suite, such as extensions.

Service Health Portal

Service Health Matrix

The service status portal provides a two-dimensional matrix view of active events mapped to a given service and geography. To help clarify which specific aspects of the service are affected, we communicate impact of each of these services by geographic region in the service matrix.

Event Logs

More information on active events are found in the form of an "event log" that's displayed on the top of the service health matrix. Each log will have additional associated information in the form of the impacted service and geography as well as the duration of a specific event. You can click through the provided hyperlink to navigate into the event log, which provides detailed information on the event being investigated.

The [Status history section](#) provides a view into current active events as well as past events. You can also filter the logs to adjust the scope of your search into past events. In addition, you can use the REST API build automated alerting solutions to help you stay on top of events with Azure DevOps services.

You can use [the RSS feed](#) to subscribe and receive information in your feed reader.

Using REST APIs

For users who are looking to build an automated solution to monitor the infrastructure incidents, we provide REST APIs to retrieve the current health status of each of the Azure DevOps services. Stay tuned for more information.

Get Help

If you're experiencing an issue with Azure DevOps and see a corresponding event communicated on the service health portal, be assured that we're working to restore normal operations of the service. You don't need to take any further action to notify us. However, if you don't see your issue reported on the Azure DevOps service health page, refer to the resources available on [Azure DevOps support page](#). That page has resources to report bugs, interact with a friendly support bot, and contact our support team for additional assistance.

Data protection overview

9/24/2019 • 25 minutes to read • [Edit Online](#)

Azure DevOps Services

Azure DevOps Services is a cloud-hosted application for your development projects, from planning through deployment. Based on the capabilities of Team Foundation Server, with additional cloud services, Azure DevOps manages your source code, work items, builds, tests, and much more. Azure DevOps uses Azure's Platform as a Service infrastructure and many of Azure's services, including Azure SQL databases, to deliver a reliable, globally available service for your development projects. Because important data is at stake, this white paper discusses the steps that Microsoft takes to keep your projects safe, available, secure, and private. In addition, it describes the role you play in keeping your projects safe and secure.

This article is part of our effort to illuminate how we manage and protect your data and is intended for organization administrators and IT professionals who manage their project assets daily. It will be most useful to individuals who are already familiar with Azure DevOps and want to know more about how Microsoft protects the assets that are stored in Azure DevOps.

Our commitment

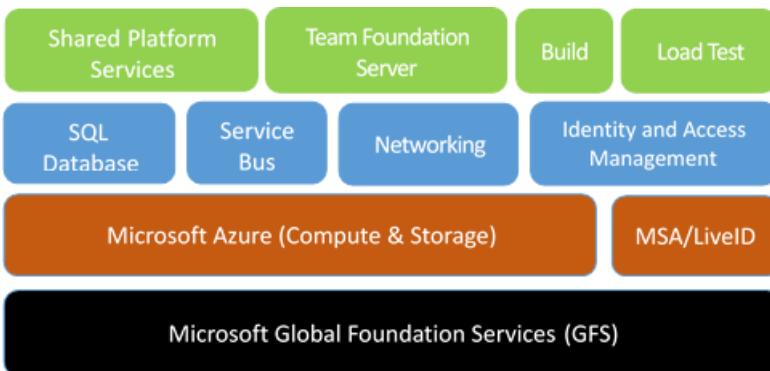
Microsoft is committed to ensuring that your projects remain safe and secure, without exception. When stored in Azure DevOps, your projects benefit from multiple layers of security and governance technologies, operational practices, and compliance policies. We enforce data privacy and integrity both at rest and in transit.

As we look at the broader landscape of threats facing Azure DevOps customers, they boil down to four basic categories: data availability, service availability, service security, and data privacy. We will investigate each of these categories to explore specific threats and explain what Azure DevOps does to address them through both the technology that we use and the way we put it into practice. However, we will first describe how data is stored and how Azure DevOps manages access to your data.

Because proper data protection also requires active engagement of customer administrators and users, we also discuss steps you should take to protect your project assets from unauthorized disclosure and tampering. Much of this has to do with being explicit about granting permissions to user access points in order to have confidence that only the right people are accessing data within Azure DevOps.

Regardless of your approach, you should consider all data potentially "at risk" no matter where it is or how it is being used; this is true for both data in the cloud as well as data stored in a private data center. Thus, it is important to classify your data, its sensitivity / risk horizon, and the damage it could do if it is compromised. You should also categorize your data relative to an overall information security management policy.

Built on Azure



Azure DevOps Services is hosted entirely in Azure data centers and uses many of the core Azure services including Compute, Storage, Networking, SQL Database, Identity and Access Management Services, and Service Bus. This lets us focus on the unique aspects of running Azure DevOps while taking advantage of the state-of-the-art capabilities, protection, and industry certifications available from the Azure platform.

Azure DevOps Services uses Azure Storage as the primary repository for service metadata and customer data. Depending on the type of data and the storage and retrieval needs, we use Azure Blob (binary large objects) storage and Azure SQL data storage. To provide a seamless experience, we work hard to abstract these details from the end user. However, to discuss the details surrounding Azure DevOps approach to data protection, some background in these elements is important.

Azure Blob storage is generally used to store large chunks of unstructured data. All projects use the Azure Blob storage service. This data includes potentially sensitive or private information such as the contents of source files and the attachments on work items. For most projects, the majority of storage in use is this type of unstructured blob storage. For more information, see documentation on [Azure Blob Storage](#).

Azure SQL database storage is used to store the structured and transactional aspects of your organization, including project metadata, the versioned source control history, and work item details. Database storage gives you fast access to the important elements of your project and provides indexes into the blob storage to look up files and attachments. For more information, see documentation on [Azure SQL Database](#).

Administrators can manage access to resources by [granting or restricting permissions](#) on user identities or groups. Azure DevOps uses federated authentication of user identities [via Azure Active Directory](#) (Azure AD) and Microsoft Accounts (MSA, formerly LiveID). During authentication, the user is routed to the authentication provider (Azure AD or MSA) where they provide their credentials. Once the authentication provider has verified the user's credentials, Azure DevOps issues an authentication cookie to the user, which allows them to remain authenticated against Azure DevOps. In this way, the user's credential information is never shared directly with Azure DevOps. For each Azure DevOps resource that the user attempts to access, permissions are validated based on the user's explicit permissions as well as permissions inherited through group membership. Administrators can leverage access controls to protect [access to organization](#), project collection, team project, and team scoped data and functionality, as well as to more specific assets like version control folders and work item area paths.

Data availability

Azure DevOps Services leverages many of the Azure storage features to ensure data availability in the case of hardware failure, service disruption, or region disaster. Additionally, the Azure DevOps team follows procedures to protect data from accidental or malicious deletion.

Data redundancy

To protect data in the case of hardware or service failures, Microsoft Azure storage geo-replicates customer data between two locations within the same region that are hundreds of miles apart; for instance, between North and West Europe or between North and South United States (except Brazil). For Azure blobs, customer data is replicated three times within a single region and is replicated asynchronously to a second region hundreds of

miles away. As such, Azure always maintains the equivalent of six copies of your data. This enables us to failover to a separate region in the case of a major outage or disaster while also providing local redundancy for hardware failures within a region. For Azure SQL database storage, daily backups are maintained offsite in the case of a regional disaster.

NOTE

Note the following regarding data redundancy and fail over:

- There is an inherent delta measured in minutes when we replicate your data between the primary and secondary region
- Fail over to the secondary region is a decision that we must make centrally as it impacts all customers on the affected scale unit. Except in extreme circumstances, we'll opt to not fail over so that customer data is not lost
- Azure DevOps offers a 99.9% uptime SLA guarantee and will refund portion of the monthly charges if we miss the SLA in a specific month
- Because there is only one region in Brazil, customer data in Brazil is replicated to South Central US for disaster recovery purposes

Mistakes happen

To protect against accidental deletion of data, either by our customers or by our operation team, we also take point-in-time backups of both the Azure blob and the SQL databases. Our approach to these backups varies based on the storage type. For blobs, we have a separate copy of all blobs and regularly append new changes to each storage account. Since this data is immutable, we don't need to rewrite any existing storage as part of our backup procedures. SQL Azure, on the other hand, handles backups as a standard part of their service which we rely on. In both cases, these backups are also replicated in a paired region to ensure we can recover from a regional outage.

In addition, we perform a "soft delete" for organization deletion operations. This lets us recover entire organizations for up to 28 days after deletion.

Practice is critical

Having multiple, redundant backups of your data is good but without practice, restoring can be unpredictable. It's been said that "backups never fail, it's the restores that do". While technically incorrect, the sentiment is right. The good news is that we regularly practice restoring various data sets from backup. The geo-redundant storage that we get from Azure is tested regularly. In addition, from time to time we restore from backups to recover from human error, such as when a customer has inadvertently deleted a project in Azure DevOps. As mentioned, we have the capability of restoring your organization's data to any point in time over the last 28 days. While our turnaround time sometimes takes more than a day, we have always been able to restore the customer's data given enough time. Since there are many permutations of disaster and data corruption scenarios, we continue to plan and execute new tests on a regular basis to ensure our systems and associated process are up to the challenge.

Service availability

Ensuring that Azure DevOps Services is available for you to access your organization and associated assets is of utmost importance to us.

DDoS protections

In some cases, a malicious distributed denial-of-service (DDoS) attack can affect service availability. Azure has a DDoS defense system that helps prevent attacks against our service. It uses standard detection and mitigation techniques such as SYN cookies, rate limiting and connection limits. The system is designed not only to withstand attacks from the outside but also from within Azure. For application-specific attacks that can penetrate the Azure defense systems, Azure DevOps establishes application and organization level quotas and throttling to prevent any overuse of key service resources during an attack or accidental misuse of resources.

Live site response

While we strive for the service to be available 100% of the time, sometimes things happen that prevent us from meeting that goal. If this happens, we provide transparency to our users throughout the incident. Our 24x7 operations team is always on hand to rapidly identify the issue and to engage the necessary development team resources. Those resources then address the problem. They also aim to update the service status page within minutes of detecting an issue that affects the service. Once the team has addressed an issue, our "live-site incident" process continues as we identify the root cause of the issue and track the necessary changes to ensure we prevent similar issues in the future.

Azure DevOps live-site management processes are crafted to ensure a deep focus on service health and customer experience. Our processes minimize our time to detect, respond to, and mitigate impacting issues. Ownership for Live site is shared across all engineering disciplines, so there are continual improvements evolving out of direct experience. This means that monitoring, diagnostics, resiliency, and quality assurance processes are improved over time. Live-site management in Azure DevOps is broken into three distinct tracks, shown as follows:

Telemetry	Incident Management	Live-site Review
<ul style="list-style-type: none">▪ Alerts –Define health alerts for failure modes▪ Diagnostics – Deliver instrumentation data and operational reports▪ Troubleshooting Guides – Guidance for investigating an alert is defined by the feature and then refined by the Service Engineer.▪ Failure Mode Testing –The Service Delivery (SD) team performs failure testing to ensure alerts fire as expected▪ Onboarding –The Feature team works with their Service Engineer (SE) to onboard new alerts to the 24 x 7 team.	<ul style="list-style-type: none">▪ Detection – Product alerts detect health issues and start the Live Site Incident (LSI) process▪ Triage – The 24 x 7 team receives all critical alerts and confirms impact using TFS guidance▪ Escalation – Both Dev and Ops have individuals in an on-call rotation. SE is initial escalation path. The SE will call Dev as needed▪ Incident Management – A bridge is managed by the SE who engages Dev. and Partners to troubleshoot▪ Resolution –Communication and service restoration are actively driven until customer impact is eliminated	<ul style="list-style-type: none">▪ Goal – Weekly review of LSI ensures that leadership has visibility into live site health and repeat issues▪ Cadence – Incident from prior week have root cause documented then reviewed on weekly basis▪ Audience – VS Leadership. Partner team when they drive impact. Developer attends to provide details on Service incident▪ Ownership -Dev. owns reviews for App and Deploy issues. SD owns for Platform issues▪ Driving Improvements – Bugs and problem work items are logged for gaps (e.g. – missing alerts) and repeat root cause

The operations team also monitors the availability metrics for individual organizations. These metrics provide insights into specific conditions that might affect only some of our customers. Investigations into this data can often result in targeted improvements to address customer-specific issues. In some cases, we will even contact the customer directly to understand their experience and work with them to improve the service from their vantage point.

We understand that availability of our service is an integral part of your team's success. Because of this, we publish a service level agreement (SLA) and provide a financial guarantee to ensure we meet this agreement each month. For more specifics on our SLA and financial guarantees, please see [SLA for Azure DevOps](#).

Sometimes our partner teams or dependencies have incidents that affect Azure DevOps. All our partner teams follow similar approaches to identifying, resolving, and learning from these service outages.

Service security

Ensuring a secure service requires constant vigilance, from proper design and coding techniques, all the way through to the way we operate the service. Along those lines, we actively invest in the prevention of security holes and in breach detection. In the event of a breach, we use security response plans to minimize data leakage, loss or corruption. To learn more about how security and identity are managed, see [About security and identity](#).

Secure by design

To implement industry best practices and stay on the forefront of information security, we engage with various teams within Microsoft including Azure, Global Foundation Services (GFS), and Trustworthy Computing. Microsoft's Security Development Lifecycle (SDL) is at the core of our development process and Microsoft's Operational Security Assurance (OSA) program guides our cloud operation procedures. The SDL and OSA methodologies address security threats throughout the development process and operation of Azure DevOps.

They specify requirements that include threat modeling during service design, following design and code best practices, verifying security with standard tooling and testing, limiting access to operational and customer data, and gating rollout of new features through a rigid approval process.

Because the security landscape is continually changing, it is important for our team to keep current with the latest in best practices. We have annual training requirements for all engineers and operations personnel working on Azure DevOps. In addition, we sponsor informal "brownbag" meetings. These meetings are hosted by our own engineers. After they've solved an issue, they share what they've learned with the rest of the team.

A cloud service is only as secure as the host platform. Azure DevOps uses Azure's Platform as a Service (PaaS) offering for much of our infrastructure. PaaS automatically provides regular updates for known security vulnerabilities. When we host virtual machines in Azure using their Infrastructure as a Service (IaaS) offering, such as for our [hosted build service](#), we regularly update those images to include the latest security patches available from Windows Update. The same update rigor applies for our on-premises machines, including those used for deployment, monitoring, and reporting.

Our team conducts regular security-focused penetration testing of Azure DevOps. Using the same techniques and mechanisms as real malicious attackers, penetration testing tries to exploit the live production services and infrastructure of Azure DevOps. The goal is to identify real-world vulnerabilities, configurations errors or other security gaps in a controlled process. The team reviews the results to identify other areas of improvement and to increase the quality of the preventative systems and training.

Credential security

Your credentials in Azure DevOps are stored using industry best practices. Learn more about [credential storage](#).

Reporting security issues

If during your penetration testing you believe you have discovered a potential security flaw related to the Azure DevOps service, please report it to Microsoft within 24 hours by following the instructions on the [Report a Computer Security Vulnerability](#) page.

IMPORTANT

While notifying Microsoft of penetration testing activities is no longer required, customers must still comply with the [Microsoft Cloud Unified Penetration Testing Rules of Engagement](#).

Azure DevOps Bug Bounty Program

Azure DevOps participates in the [Microsoft Cloud Bounty Program](#) to reward security researchers who report issues to us, and to encourage more people to help us keep Azure DevOps secure. Please visit the [Azure DevOps Bounty Program](#) page for more details.

Restricting access

We maintain strict control over who has access to our production environment and customer data. Access is only granted at the level of least privilege required and only after proper justifications are provided and verified. If a team member needs access to resolve an urgent issue or deploy a configuration change, they must apply for "just in time" access to the production service. Access is revoked as soon as the situation is resolved. Access requests and approvals are tracked and monitored in a separate system. All access to the system is correlated against these approvals and if unapproved access is detected, an alert is raised for the operations team to investigate.

If the username and password for one of our developers or operation staff were ever stolen, data is still protected because we use two-factor authentication for all remote system access. This means that additional authentication checks via smart card or a phone call to a pre-approved number must take place before any remote access to the service is permitted.

In addition, secrets that we use to manage and maintain the service, such as RDP passwords, SSL certificates, and encryption keys are managed, stored, and transmitted securely through the Azure Management Portal. Any

access to these secrets requires specific permission, which is logged and recorded in a secure manner. All secrets are rotated on a regular cadence and can be rotated on-demand in the case of a security event.

The Azure DevOps operations team uses hardened administrator workstations to manage the service. These machines run a minimal number of applications and operate in a logically segmented environment. Operations team members must provide specific credentials with two-factor authentication to access the workstations and all access is monitored and securely logged. To isolate the service from outside tampering, applications such as Outlook and Office, which are often targets of spear-phishing and other types of attacks, are not permitted in this environment.

Intrusion protection & response

To ensure data is not intercepted or modified while in transit between you and Azure DevOps, we encrypt via HTTPS / SSL.

In addition, data we store on your behalf in Azure DevOps is encrypted as follows:

- For data stored in Azure SQL databases, Azure DevOps adopted [Transparent Data Encryption \(TDE\)](#) to protect against the threat of malicious activity by performing real-time encryption of the database, associated backups, and transaction log files at rest.
- Azure Blob Storage connections are encrypted to protect your data in transit. To protect data at rest stored in our Azure Blob Storage, we have adopted [Azure Storage Service Encryption \(SSE\)](#).

To learn more about how we encrypt your data, please visit the following [blog post](#).

To ensure that activities within the service are legitimate, as well as to detect breaches or attempted breaches, we leverage Azure's infrastructure to log and monitor key aspects of the service. In addition, all deployment and administrator activities are securely logged, as is operator access to production storage. Real-time alerts are raised because the log information is automatically analyzed to uncover potentially malicious or unauthorized behavior.

In the case where a possible intrusion has been detected or high priority security vulnerability has been identified, we have a clear security incident response plan. This plan outlines responsible parties, steps required to secure customer data, and how to engage with security experts in Microsoft Security Response Center (MSRC), Global Foundation Services (GFS), Azure and members of the Azure DevOps leadership team. We will also notify any organization owners if we believe their data was disclosed or corrupted so that they can take appropriate steps to remedy the situation.

Finally, to help combat emerging threats, we employ an Assume Breach strategy. A highly specialized group of security experts within Microsoft, known as the Red Team, assumes the role of sophisticated adversaries. This team tests our breach detection and response, enabling us to accurately measure our readiness and the impacts of real-world attacks. This strategy strengthens threat detection, response, and defense of the service. It also allows us to validate and improve the effectiveness of our entire security program.

Data privacy

We want you to have confidence that your data is being handled appropriately and for legitimate uses. Part of that assurance involves appropriately restricting usage so that your data is used only for legitimate reasons.

The General Data Protection Regulation (GDPR)

The General Data Protection Regulation (GDPR) is the biggest change in data protection laws in Europe since the 1995 introduction of the European Union (EU) Data Protection Directive 95/46/EC. The GDPR's main objective is to strengthen the protection and security of your personal data and replaced the Directive and all local laws relating to it. Azure DevOps is relied upon as a system of record with strict integrity, traceability, and audit rules. We view all information within Azure DevOps to be business critical and therefore cannot be modified from its original state. These existing obligations affect our delete and retention obligations for GDPR. As such, we do not support GDPR delete requests from within Azure DevOps. We have ensured that when an entire organization is

deleted that all associated data and telemetry about that organization and its members are removed from our system (after the requisite 30-day soft-delete period).

To learn more about how Azure DevOps honors Data Subject Requests (DSR), please visit the following [page](#). To learn more about the GDPR regulation, please visit the following page in [Microsoft's Trust Center](#).

Data residency and sovereignty

We know our customers care deeply about data security and privacy. Azure DevOps is available in the following 8 geographies across the world: United States, Canada, Europe, UK, India, Australia, Asia Pacific, and Brazil. While we default your organization to your closest region, you have the option to choose a different region. If you change your mind later, our CSS team can help you migrate your organization to a different region. Azure DevOps will not move or replicate customer data outside of the chosen geography. Our backup procedures geo-replicate customer data between a second region within the same geography except for organizations located in Brazil, these are replicated to South Central US for disaster recovery purposes.

To learn more about data location, see [Azure DevOps data location](#).

Law enforcement access

In some cases, third parties such as law enforcement entities may approach us to obtain access to customer data stored within Azure DevOps. By policy, we will attempt to redirect the requests to the organization Owner for resolution. When we are compelled by court order to disclose customer data to a third party, we will make a reasonable effort to notify the organization Owner in advance unless we are legally prohibited from doing so.

Some customers require that their data be stored in a particular geographic location to ensure a specific legal jurisdiction for any law enforcement activities. At this time, Azure DevOps can host organizations in 8 geographies across the world: United States, Canada, Europe, UK, India, Australia, Asia Pacific, and Brazil. All customer data such as source code, work items, test results as well as the geo-redundant mirrors and offsite backups are maintained within the selected geography.

Microsoft access

From time to time, Microsoft employees need to obtain access to customer data stored within Azure DevOps. As a precaution, all employees who have or may ever have access to customer data must pass a background check, which verifies previous employment and criminal convictions. In addition, we permit access to the production systems only when there's a live site incident or other approved maintenance activity, which is logged and monitored.

Since not all data within our system is treated the same, data is classified to distinguish between customer data (what you upload to Azure DevOps), organization data (information used when signing up for or administering your organization), and Microsoft data (information required for or collected through the operation of the service). Based on the classification we control usage scenarios, geolocation requirements, access restrictions and retention requirements.

Microsoft promotional use

From time to time, we want to contact customers to let them know about additional features and services that might be useful. Since not all customers want to be contacted about these offers, we allow you to opt-in and opt-out of marketing email communications. We never use customer data to target specific offers for specific users or organizations. Instead, we leverage organization data and aggregate usage statistics at the organization level to determine groups of organizations that should receive specific offers.

Building confidence

In addition to these protections, we have also taken steps outside of the service itself to help you decide to move forward with Azure DevOps. These include Microsoft's own internal adoption policies, the level of transparency that we provide into the state of our service, and our progress towards receiving certification of our information security management systems. All these efforts are designed to build your confidence in Azure DevOps.

Internal adoption

Teams across Microsoft have begun adopting Azure DevOps internally. The Azure DevOps team moved into an organization in 2014 and uses it extensively. More broadly, we have established guidelines to enable the adoption plans for other teams. Obviously, large teams move more gradually than smaller ones, given their investments in existing DevOps systems. For teams able to move quickly, we have established a project classification approach. It assesses our risk tolerance, based on project characteristics, to determine if the project is appropriate for Azure DevOps. For larger teams, the adoption typically occurs in phases with more planning. Additional requirements for internal projects include associating the organization with the Microsoft.com Azure Active Directory to ensure proper user identity lifecycle and password complexity along with requiring the use of two-factor authentication for more sensitive projects.

Transparency

We are convinced that transparency around how we design and operate our service is critical to establishing trust with our customers. This white paper is part of our effort to shed light on how we manage and protect your data. In addition, we maintain a [blog](#) that provides real-time updates whenever a service disruption, planned or unplanned, takes place so you can adjust your activities as needed. Furthermore, Brian Harry, the corporate vice-president in charge of Azure DevOps, maintains an active [blog](#) addressing, among other things, lessons learned by operating the service.

Compliance certifications

For some customers, it is important to understand third-party evaluation of our data security procedures. Towards that end, we have achieved ISO 27001:2013, HIPAA (Health Insurance Portability and Accountability Act) BAA (Business Associate Agreement), EU Model Clauses, SOC 1 Type 2 and SOC 2 Type 2 certifications. The SOC audit for Azure DevOps covers controls for data security, availability, processing integrity, and confidentiality. Azure DevOps' SOC reports are available via the [Microsoft's Service Trust Portal](#). If you don't have access to Microsoft's Service Trust Portal, you can contact [Azure DevOps ServicesSOCReports](@mailto:Azure DevOps ServicesSOCReports@microsoft.com) to request a copy of Azure DevOps' SOC Reports.

Steps you can take

Proper data protection requires active engagement of customer administrators and users. Your project data stored within Azure DevOps is only as secure as the end user access points. So, it is important to match the level of permission strictness and granularity for those organizations with the level of sensitivity of your project.

Classify your data

The first step is to classify your data based on its sensitivity / risk horizon, and the damage that could occur if it is compromised. Many enterprises have existing classification methods that can be reused when projects move to Azure DevOps. Refer to these [materials](#) for more information on how to classify your data.

Adopt Azure Active Directory

Another action you can take to improve the security of your end users' credentials is to use Azure Active Directory (Azure AD) instead of Microsoft Accounts (MSA) to manage your organization's access to Azure DevOps. This allows your IT department to manage its end user access policy including password complexity, password refreshes and expiration if the user leaves your organization. Through Active Directory federation, you can directly link Azure Active Directory to your organization's central directory, so you have only one location to manage these details for your enterprise. See the following brief comparison between MSA and Azure AD characteristics relative to Azure DevOps access:

PROPERTIES	MSA	AZURE AD
Identity creator	User	Organization

Properties	MSA	Azure AD
Single user name / password for all work assets	No	Yes
Password lifetime & complexity control	User	Organization
Azure DevOps membership limits	Any MSA	Organization's directory
Traceable identity	No	Yes
Organization & IP ownership	Unclear	Organization
2-factor authentication enrollment	User	Organization
Device-based conditional access	No	Organization

You can learn more about how to [configure this support for your organization](#).

Require two-factor authentication

In some cases, you might want to restrict access to your organization by requiring more than one factor to sign in. Azure AD lets you require multiple factors, such as phone authentication in addition to a username and password, for all authentication requests. You can [learn more](#) about turning on multifactor authentication for Azure AD.

Use BitLocker

For sensitive projects, we also recommend use of BitLocker on your Windows laptop or desktop computer. BitLocker encrypts the entire drive on which Windows and your data reside, keeping everything safe. Once BitLocker is enabled, it automatically encrypts any file you save on that drive. If your laptop or desktop machine were to fall into the wrong hands, BitLocker prevents unauthorized access of local copies of data from your projects.

Limit use of Alternate Authentication Credentials

The default authentication mechanism for Git related tooling is alternate authentication (sometimes referred to as Basic Authentication). This mechanism allows the end user to set up an alternate username and password for use during Git command line operations. This username and password combination can also be used to access any other data for which that user has permissions. By its nature, alternate authentication credentials are less secure than the default federated authentication. However, we have taken steps to help you make secure choices. For example, all communication is sent over HTTPS and there are password complexity requirements. Nevertheless, your organization should evaluate if additional policies are required to meet your project security requirements. You can [learn more](#) about disabling alternate authentication credentials altogether for your organization if it doesn't meet your security requirements.

Secure access to your organization

Azure Active Directory (Azure AD) provides the capability for administrators to control access to Azure resources and applications such as Azure DevOps. With conditional access control in place, Azure AD checks for the specific conditions you set for a user to access an application. After access requirements are met, the user is authenticated and can access the application. Visit the [Azure documentation site](#) to learn more about conditional access policy (CAP). Azure DevOps now enforces conditional access policies for custom Azure DevOps authentication mechanisms including personal access tokens (PATs), alternate authentication, OAuth and SSH keys. If accessing Azure DevOps through a third party client, like git.exe, only IP based policies will be honored; any other policy will automatically fail as the client doesn't pass the necessary information to validate the policy. For example, if a policy requires MFA and the client can't support MFA, the policy will fail and the user will automatically be blocked.

Additional resources

In addition to this white paper, there are other resources available for your review and education. These include:

- [Azure DevOps home page](#)
- [Azure DevOps status](#)
- [Azure DevOps credential storage](#)
- [Azure DevOps data location](#)
- [Developer Services privacy statement](#)
- [Azure DevOps support](#)
- [Developer Services Agreement](#)
- [Azure trust center](#)
- [Microsoft Security Development Lifecycle](#)
- [Create and revoke your PATs](#)
- [Revoke user PATs - for admins](#)
- [Token expiration](#)

(c) 2018 Microsoft Corporation. All rights reserved. This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site references, may change without notice. You bear the risk of using it.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes.

Data locations for Azure DevOps

9/24/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

Azure DevOps operates in the geographical locations ("geos") listed below. To determine where customer data is stored, you can choose the location of the organization during initial sign up and creation of the organization.

Data locations

Your data is stored within the following locations:

- Australia
- Brazil
- Canada
- East Asia
- Europe
- India
- United Kingdom
- United States

Azure DevOps stores information that is global in nature, such as user identities and profile information, in a data center located in the United States. All customer data, such as source code, work items, and test results, as well as the geo-redundant mirrors and offsite backups, are maintained within the selected geography.

NOTE

Because there is only one region in Brazil, customer data is replicated to south-central United States for disaster recovery and load balancing purposes. For more information, see the [Azure data center map](#).

For builds and releases configured to run on Microsoft-provided macOS agents, Azure DevOps stores associated customer data in the United States in a data center that is owned and managed by a third party with reduced information security certification assurances.

Azure DevOps works with and uses many Microsoft Azure services. For details on customer data retention by location, see the [Azure data center map](#).

Transferring your data

Microsoft does not transfer customer data outside the selected region, except when it is necessary for Microsoft to provide customer support, troubleshoot the service, or comply with legal requirements. In such a case, you configure an organization to enable such transfer of your data using preview, beta, or other pre-release services, which typically store your data in the United States, but may store it globally.

NOTE

Microsoft does not control or limit the regions from which you or your users may access your data.

Related articles

- [Get started with Azure DevOps](#)
- [Data protection overview](#)

How we store your credentials for Azure DevOps Services

9/24/2019 • 2 minutes to read • [Edit Online](#)

Azure DevOps Services

Credential security

Microsoft is committed to ensuring that your projects remain safe and secure, without exception. In Azure DevOps, your projects benefit from multiple layers of security and governance technologies, operational practices, and compliance policies. We enforce data privacy and integrity both at rest and in transit. In addition, we adhere to the following practices with respect to the credentials or secrets that Azure DevOps stores. To learn more about how to choose the right authentication mechanism, see [Guidance for authentication](#).

Alternate credentials (basic auth)

- We store two values, a 16-byte password salt and a 32-byte password hash
- Raw password is provided directly by the caller over SSL
- Password salt is randomly generated in-memory on the server side using RNGCryptoServiceProvider each time a password is created or changed
- Password hash is generated in-memory on the server side from the raw password and password salt bytes using Rfc2898DeriveBytes with 1000 iterations
- Salt and hash are stored in a database

Personal access tokens (PATs)

- We store a hash of the PAT
- Raw PAT is generated in-memory on the server side as 32 bytes randomly generated through RNGCryptoServiceProvider then shared with the caller as a base-32-encoded string. This value is NOT stored.
- PAT hash is generated in-memory on the server side as an *HMACSHA256Hash* of the raw PAT using a 64-byte symmetric signing key stored in our key vault
- Hash is stored in our database

Secure shell (SSH) keys

- We store a hash of the enclosing organization ID and the SSH public key
- Raw public key is provided directly by the caller over SSL
- SSH hash is generated in-memory on the server side as an *HMACSHA256Hash* of the organization ID and raw public key using a 64-byte symmetric signing key stored in our key vault
- Hash is stored in our database

OAuth credentials (JWTs)

- These are issued as fully self-describing JSON web tokens (JWTs) and are NOT stored in our service
- The claims in JWTs issued and presented to our service are validated using a certificate stored in our key vault