

# Contents

[Docs contributor guide](#)

[Contribute to this guide](#)

[What's new in this guide](#)

[Get started](#)

[Get started options](#)

[Get started checklist](#)

[Set up your GitHub account](#)

[Make minor edits](#)

[Make major updates](#)

[Install Git/Markdown tools](#)

[Fork and clone a repository](#)

[Add a new article](#)

[Make changes to an article](#)

[Create a pull request](#)

[Publishing best practices](#)

[Help options](#)

[Partner quick reference](#)

[Content](#)

[Plan your content](#)

[Content process](#)

[Content process overview](#)

[Planning content](#)

[Designing content](#)

[Writing content](#)

[Reviewing and publishing content](#)

[Maintaining content](#)

[Topic types](#)

[Choose a topic type](#)

[Write an overview](#)

[Write a quickstart](#)

[Write a quickstart](#)

[Write an SDK library quickstart](#)

[Write an ARM template quickstart](#)

[Write a tutorial](#)

[Write a concept](#)

[Write a how-to guide](#)

[Write FAQ content](#)

[FAQ best practices](#)

[Write a structured FAQ](#)

[Add zone pivots to articles](#)

[Write a region move article](#)

[Write about load balancing](#)

[Write about managing costs](#)

[Write about Azure RBAC](#)

[Write about Azure monitoring](#)

[Write about public container images](#)

[Write troubleshooting content](#)

[Troubleshoot overview](#)

[Choose a format](#)

[Write general troubleshooting](#)

[Write known issues](#)

[Write a problem resolution](#)

[Write an error code list](#)

[Write a Security Baseline](#)

[Write about "best practices"](#)

[Write legal content](#)

[Write legal content](#)

[Legal guidelines](#)

[Set up code samples](#)

[Code samples overview](#)

[Concepts](#)

[Get Started](#)

[Basics](#)

[Guidance & Governance](#)

[README Template](#)

[Process](#)

[Onboarding](#)

[Diagnostics](#)

[Platform-Specific Guidance](#)

[Get Started](#)

[Java](#)

[JavaScript](#)

[Templates](#)

[Find your template](#)

[Overview template](#)

[Quickstart template](#)

[Quickstart - ARM template](#)

[Tutorial template](#)

[Concept template](#)

[How-to guide template](#)

[Create or update a template](#)

[Azure templates](#)

[Tutorial](#)

[Manage costs](#)

[Monitoring](#)

[.NET templates](#)

[Tutorial](#)

[Advanced tutorial](#)

[Quickstart](#)

[SQL templates](#)

[Tutorial](#)

[Troubleshoot templates](#)

[General troubleshooting](#)

- [Known issues](#)
- [Problem resolution](#)
- [What \(not to\) to publish](#)
- [Respond to inappropriate content](#)
- [Content channel guidance](#)
  - [Case studies + customer stories](#)
  - [White papers](#)
  - [Blogs](#)
  - [Downloadable content](#)
- [Optimize for search](#)
  - [Get started with SEO](#)
  - [Basics - SEO cheat sheet](#)
  - [Checklist for web-friendly pages](#)
  - [Writing examples](#)
    - [Write SEO-friendly titles](#)
    - [Write SEO-friendly meta descriptions](#)
    - [Write SEO-friendly H1s](#)
    - [Rank in the Google quick answer box](#)
  - [Keyword data and research](#)
    - [Get keywords and search rank for docs](#)
    - [Keyword research and tracking](#)
  - [File name and folder path guidelines](#)
  - [Troubleshoot SEO issues](#)
  - [Run an SEO hack](#)
- [Freshness + maintenance](#)
  - [Delete, rename, archive, or move articles](#)
    - [Archive, retire, and removal policy](#)
    - [Retire or rename an article](#)
    - [Archive and content removal guidelines](#)
    - [Archive/retire articles using Docs portal](#)
    - [Archive entire docset](#)
    - [Refactor, rename, or move many articles](#)

[Rename a product or service in content](#)

[Redirect obsolete articles](#)

[Use multiple redirect files](#)

[Archive \(previous-versions\) overview](#)

## [Art/multimedia](#)

[Media gallery guidance](#)

[Add art to articles](#)

[Create a screenshot](#)

[Create an expandable screenshot](#)

[Create conceptual art](#)

[Create an animated GIF](#)

[Create videos](#)

[Create and publish a video](#)

[Create diagrams](#)

[Diagrams for technical communicators](#)

[How to choose a diagram type](#)

[How to draw a diagram type](#)

[How to draw from scratch](#)

[How to provide feedback](#)

[Diagram cookbook](#)

[Process mapping](#)

[Process mapping overview](#)

[Process definition](#)

[Process summary](#)

[Process map](#)

[Diagrams as code](#)

[About diagrams](#)

## [Writing](#)

[Style guidelines](#)

[Technical principles](#)

[Technical principles checklist](#)

[Azure CLI & Azure PowerShell - best practices](#)

- [Azure CLI article guidelines](#)
- [Azure CLI editor's checklist](#)
- [Azure PowerShell code conventions](#)
- [Request a Technical Focused Review \(TFR\)](#)
- [Writing principles](#)
  - [Writing voice](#)
  - [Writing principles checklist](#)
  - [Writing principles job aid](#)
  - [Everyday word list](#)
  - [Request a Customer Focused Review \(CFR\)](#)
- [Reference content](#)
  - [Reference content](#)
    - [SDK and reference overview](#)
    - [Reference documentation overview](#)
    - [Reference content FAQ](#)
    - [What is Swagger](#)
  - [How to](#)
    - [Document SDK client libraries](#)
    - [How to write .NET docs](#)
    - [How to write CLI docs](#)
    - [How to write Powershell docs](#)
  - [Resources](#)
    - [Swagger newbie guide](#)
    - [Template - Library index](#)
    - [Azure reference style guide](#)
    - [Azure SDK onboarding](#)
  - [Unified content model](#)
  - [Unified content model FAQ](#)
- [Navigation](#)
  - [Breadcrumbs guidelines](#)
  - [TOCs](#)
    - [TOC structure](#)

- [TOC file format](#)
- [Contextual TOC + breadcrumbs](#)
- [TOC Visualizer](#)
- [Hub/landing pages](#)
  - [Create or update a hub page](#)
  - [Create or update a landing page](#)
- [Navigation](#)
  - [Global navigation governance](#)
  - [Product-level navigation guidelines](#)
- [Releases](#)
  - [Plan content releases](#)
  - [New contributor content checklist](#)
  - [How content resources are assigned](#)
  - [Using release branches](#)
  - [Preview content](#)
  - [Request approval for an exception or experiment](#)
  - [Public vs private repos](#)
  - [Editorial service](#)
  - [Publish to air-gapped clouds](#)
- [Metadata](#)
  - [Metadata overview](#)
  - [How to apply metadata](#)
  - [Metadata tagging](#)
  - [Metadata attributes](#)
  - [APIScan metadata](#)
  - [Metadata changes](#)
  - [Find the author of an article via metadata](#)
  - [Metadata lists \(allowlists\)](#)
  - [Taxonomy lists](#)
- [Markdown](#)
  - [Markdown reference](#)
  - [Add code to articles](#)

- [Add links to articles](#)
- [Link to articles from UI](#)
- [Add lists and bullets in articles](#)
- [Include reusable content in articles](#)
- [Overview of Tabbed conceptual](#)
- [SMS Markdown conversion](#)
- [Format text](#)
- [Git/GitHub](#)
  - [GitHub workflow for MicrosoftDocs](#)
  - [Migrate GitHub project documentation to Docs](#)
  - [Rebase and squash](#)
  - [Writing/Publishing best practices](#)
  - [Use GitHub for content reviews](#)
  - [GitHub glossary](#)
  - [Recover your GitHub account](#)
  - [Issues review process](#)
    - [Triage GitHub issues](#)
    - [Resolving stale GitHub issues](#)
  - [Pull request review process](#)
    - [PR comment automation](#)
    - [PR content quality criteria](#)
    - [PRMerger overview](#)
    - [Review PRs without PRMerger](#)
    - [Azure PowerShell & Azure CLI samples PR review process](#)
    - [PR submission best practices](#)
    - [Resolve a stuck pull request](#)
    - [Resolve merge conflicts](#)
    - [Break up large PRs](#)
    - [Public PR response guidance](#)
    - [Standard, out-of-band, and hotfix publishing](#)
    - [PR review and publishing schedule](#)
  - [Troubleshooting Git errors](#)

## Data, BI, & Reports

[Data and BI overview](#)

[Data advocacy \(CDAG\)](#)

[Metrics definitions](#)

[Reports and tools](#)

[Content performance report](#)

[Content health report](#)

[Documentation and reference report](#)

[Verbatims report](#)

[Github Issues report](#)

[Microsoft Docs Metrics tool \(MDM\)](#)

[Contributor Insights](#)

[Page-level trends & drill-down dashboards](#)

[Zero and low page views dashboard](#)

[Doc Subset dashboard](#)

[Troubleshoot lower-performing articles](#)

[Content performance reporting principles](#)

[Known issues + release notes](#)

[Other data/BI resources](#)

[How to connect to Kusto data clusters](#)

[Content performance canonical functions](#)

[Kusto community query library](#)

[Adobe Analytics](#)

[View and act on verbatim feedback](#)

[Request process for dashboards](#)

[Data on API documentation](#)

## Customer research

[How to do customer research](#)

[How to select a research method](#)

[How to get a UserTesting license](#)

[How to do customer interviews](#)

[How to do surveys](#)

## [Customer research FAQs](#)

[Customer feedback - Azure Pulse program](#)

[Customer feedback - PSAT program](#)

[Receive GitHub issues and public PR reports](#)

[Host a hackadoc](#)

[Run experiments](#)

## [Tools](#)

[Docs Build validation](#)

[Docs Build validation overview](#)

[PR only validation rules](#)

[Docs Build validation reference](#)

[alt-text-bad-length](#)

[alt-text-bad-value](#)

[alt-text-missing](#)

[author-missing](#)

[author-not-found](#)

[bookmark-not-found](#)

[code-block-indented](#)

[code-block-unclosed](#)

[column-header-missing](#)

[dependent-repo-branch-mismatch](#)

[deprecated-attribute](#)

[description-bad-length](#)

[description-missing](#)

[dev-language-invalid](#)

[dev-language-missing](#)

[disallowed-heading](#)

[docs-link-absolute](#)

[duplicate-alt-text](#)

[duplicate-descriptions](#)

[duplicate-h1s](#)

[duplicate-h2s](#)

[duplicate-pivot-groups](#)

[duplicate-pivot-ids](#)

[duplicate-titles](#)

[duplicate-toc-entries](#)

[duplicate-uids](#)

[file-not-found](#)

[forbidden-content](#)

[forbidden-link](#)

[h1-empty](#)

[h1-in-include](#)

[h1-in-moniker](#)

[h1-missing](#)

[h1-no-space](#)

[h1-not-first](#)

[hard-coded-locale](#)

[heading-empty](#)

[heading-no-content](#)

[heading-skipped](#)

[heading-with-underline](#)

[html-link](#)

[image-invalid](#)

[include-not-found](#)

[insecure-link](#)

[invalid-value](#)

[languages-missing](#)

[link-out-of-scope](#)

[link-redirected](#)

[link-text-missing](#)

[invalid-tab-group](#)

[manager-invalid](#)

[manager-missing](#)

[merge-conflict](#)

module-achievement-not-badge  
module-achievement-not-found  
module-badge-missing  
module-child-not-found  
module-child-not-unit  
module-uid-missing  
module-units-missing  
ms-author-missing  
ms-author-invalid  
ms-date-invalid  
ms-date-missing  
ms-prod-and-service  
ms-prod-missing  
ms-prod-or-service-missing  
ms-prod-technology-invalid  
ms-service-missing  
ms-service-subservice-invalid  
ms-topic-invalid  
ms-topic-missing  
multiple-h1s  
multiple-values  
notebook-dependent-schema-failed  
path-achievement-not-found  
path-achievement-not-trophy  
path-child-not-found  
path-child-not-module  
path-duplication  
path-modules-missing  
path-too-deep  
path-trophy-missing  
path-uid-missing  
pivot-definition-conflict

[pivot-definition-not-found](#)  
[pivot-group-missing](#)  
[pivot-group-not-found](#)  
[pivot-id-not-found](#)  
[pivot-id-unused](#)  
[preserve-view-not-set](#)  
[quiz-multiple-answers](#)  
[quiz-no-answer](#)  
[redirect-url-invalid](#)  
[redirected-file-not-removed](#)  
[rule-override-invalid](#)  
[sandbox-not-mapped](#)  
[sensitive-language](#)  
[tabbed conceptual validation](#)  
[table-syntax-invalid](#)  
[task-type-invalid](#)  
[title-missing](#)  
[title-suffix-missing](#)  
[toc-breadcrumb-link-external](#)  
[toc-missing](#)  
[token-not-found](#)  
[unit-duration-missing](#)  
[unit-multiple-parents](#)  
[unit-no-module-parent](#)  
[unit-task-and-quiz](#)  
[unit-uid-missing](#)  
[validation-timeout](#)  
[xref-not-found](#)

Error documentation not available

Docs local validation

Broken link report

CATS deprecation

[Docs Image Sorter \(prototype\)](#)

[Archive tool](#)

[Use CodeFlow for content reviews](#)

[Acrolinx](#)

[Acrolinx overview](#)

[Acrolinx in Visual Studio Code](#)

[Acrolinx minimum score](#)

[Acrolinx reporting](#)

[Acrolinx troubleshooting](#)

[AutoHotkey](#)

[Twitter App](#)

[Banner promotion request](#)

[Docs Authoring pack](#)

[Docs Authoring pack overview](#)

[Features](#)

[Dev lang completion](#)

[Image compression](#)

[Metadata explorer](#)

[Metadata updates](#)

[Reformat Markdown tables](#)

[Smart quote replacement](#)

[Sort redirects](#)

[Sort selection](#)

[Jupyter notebooks](#)

[Repo cleanup tool](#)

[SpineEdit](#)

[Localization](#)

[Global Experiences Content Localization team](#)

[Localization FAQ](#)

[Checklist for localization](#)

[Avoid force push](#)

[Guidance for legal articles](#)

- How the loc team selects content to localize
- Source escalation
- Writing for translation
- Provide Feedback to DevRel Localization Team
- How to Avoid Over-Localization
- How to specify the localization scope for an image
- Video Caption Localization
  - Request Video Caption Localization
  - Guidelines for efficient English video caption
- Repo-specific guidance
  - Azure
    - Aliases for Azure tech content
    - Doc feedback/GitHub issues process - Azure CXP
    - Doc feedback/GitHub issues process - Azure authors
    - Update Azure docs hub page
    - Request a change on azure.microsoft.com (ACOM)
    - Service updates
    - Subscriptions for writers
    - Sovereign clouds
    - New service checklist
    - Reports
      - Content Standards report
      - Use the Articles tab
      - Use the Charts tab
    - Security content in a service doc set
    - Service documentation
    - Templates
    - Security features
  - Azure Architecture Center
    - Azure Architecture Center overview
    - AAC contributions
    - Azure Architecture Center browser

Azure Architecture Center 2021 Refresh

Reference architectures

Example workloads

Solution ideas

Design patterns

Topic landing pages

AAC Markdown templates

    Design pattern template

    Reference architecture templates

    Example workload templates

    Solution idea templates

    Deprecated templates

Cloud Adoption Framework

    CAF content review process

Azure DevOps

    Track content contributions

    Content review checklist

    Azure DevOps aliases

    Repositories, docsets & owners

    Metadata guidance

    Publishing workflow

    Versioning guidance

    Tabbed conceptual tags

    Terminology & branding

    Navigation guidelines

Azure Databricks

    Publish Databricks content

    Update Databricks content

Partner Center

    Repos and publishing for Partner Center

    Use appropriate roles

    Write a Partner Center announcement

[Power BI](#)

[Power BI docs](#)

[SQL](#)

[Contribute to SQL docs](#)

[SQL docs guidance](#)

[SQL versioning guidance](#)

[SQL servicing updates](#)

[ILDC PR review and publishing guide](#)

[Visual Studio](#)

[Contribute to Visual Studio docs](#)

[Visual Studio docs guidance](#)

[Publish preview docs](#)

[Xamarin](#)

[Xamarin docs](#)

[Windows](#)

[Meriwether diff reporting tool](#)

[Resources](#)

[How to use Docs documentation](#)

[C+L office hours](#)

[Git/GitHub](#)

[Git: Git basics](#)

[Git: Pro Git e-book \(PDF\)](#)

[Git: How Git Works course at Pluralsight](#)

[GitHub: Resources to learn Git](#)

[GitHub: Git Cheat sheets](#)

[GitHub: GitHub Guides](#)

[GitHub: GitHub learning resources](#)

[GitHub: Glossary](#)

[Quality and vendor programs](#)

[Vendor supported quality program](#)

[Tutorial review vendor checklist](#)

[Public repo PR review](#)

[Vendor content authoring managed service engagement](#)

[Vendor content authoring skillsets](#)

[Vendor schedule of ongoing quality tasks](#)

[Azure top 150 freshness](#)

[Tiers, Acrolinx, and PR review coverage](#)

[Writing and publishing](#)

[Contributors Guide Cheatsheet](#)

[Best practices for Contributors](#)

[VS Code keyboard shortcuts](#)

[Understand repo roles and responsibilities](#)

[C+L business approvers and repos list](#)

[Azure-specific business approvers list](#)

[Cheat sheet: command-line steps](#)

[Create and publish sample code](#)

[Docs onboarding guide](#)

[Cloud style guide](#)

[Microsoft style guide](#)

[OPS/Docs SiteHelp portal](#)

[OPS docset portal](#)

[OPS service health portal](#)

[Content & Learning Content Production Service](#)

[Accessibility](#)

[Accessibility guidelines](#)

[Accessible multimedia](#)

[GDPR references in documentation](#)

[Terminology and inclusive language](#)

[Legal content](#)

[Policheck for GitHub](#)

[Contributor guides](#)

[Operating principles](#)

[Content standards](#)

[Available contributor guides](#)

# Contribute to the contributor guide

5/26/2021 • 4 minutes to read

The Docs contributor guide is open to contributions, but there's a process for doing so. The process exists to keep the guidance focused, well-organized, and helpful. To make the Docs contributor guide a consistent experience, we follow these [operating principles](#).

The contributor guide provides instruction for customer documentation and training. As such, it should serve as a model for customer content. The information in this guide should follow ALL our content guidance and represent the best of its kind.

## IMPORTANT

Pull requests in the docs-help-pr repository must be free of build suggestions/warnings/errors. Build validation updates can result in issues not related to your specific changes. These need to be addressed before the PR can be merged. If you need assistance mention the contact(s) noted in the **Need help?** section of the pull request.

Before you create a PR, search the [contributor guide](#) to determine if content exists for the subject you plan to cover.

## NOTE

If you need a PR merged for a deadline notify the PR reviewer at least 2 business days prior.

## Add an article

New articles should help people write, manage, and publish content on Docs. Content in the guide should also follow our [writing principles](#).

To add a new article to the Docs contributor guide:

1. Attend the next weekly Contributor guide workgroup meeting to discuss the article. Others in that group may be aware of similar content.  
Contact [Jim Parker](#) for an invite to the next session.
2. Create the pull request in the docs-help-pr repository and include reviewers that you want feedback from.
3. When the reviews are completed and your changes are ready to merge, comment with `#sign-off`. That comment changes the PR label and signals to the PR review team that the PR is ready. The PR review team will do a criteria review and either let you know of issues that need to be addressed, or approve the PR (required) and merge.

## IMPORTANT

If you're linking to examples in your new article, make sure those examples comply with our content guidance. Using examples that illustrate your point, but conflict with other guidance confuses users about the correct way to create and format content.

# Update an article

To update an existing article:

1. Make your updates and create a pull request in the docs-help-pr repository.
2. Add the author of the article as a reviewer. Authors are required to approve changes.
3. When the reviews are completed and your changes are ready to merge, comment with `#sign-off`. That comment changes the PR label and signals to the PR review team that the PR is ready. The PR review team will do a criteria review and either let you know of issues that need to be addressed, or approve the PR (required) and merge.

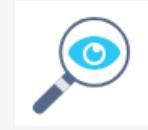
## TIP

If your proposed changes are significant, discuss those changes with the author before you make them.

## Add the "In review" tag

New articles, or updates that alter an approved process need the "In review" tag added to the top of the article.

## NOTE



THIS DOCUMENT IS IN REVIEW AND IS NOT YET SUPPORTED IN THE CONTENT STANDARDS FOR [DOCS.MICROSOFT.COM](#). WE ENCOURAGE YOU TO USE THE GUIDANCE AND PROVIDE FEEDBACK [IN OUR TWO-QUESTION SURVEY](#).

Add the following include just below the H1 in your new, or updated, article.

```
[!INCLUDE [in-review](../../includes/in-review.md)]
```

Once the article or changes are published, you're responsible for addressing feedback to improve the article. The label includes a link to a [form for collecting feedback](#). The contributor guide team will notify you if feedback is submitted. If there isn't any feedback in 60 days, the label is removed. If there is feedback and the article is not updated within 60 days, it's removed from the guide until the feedback is addressed.

## Update the TOC

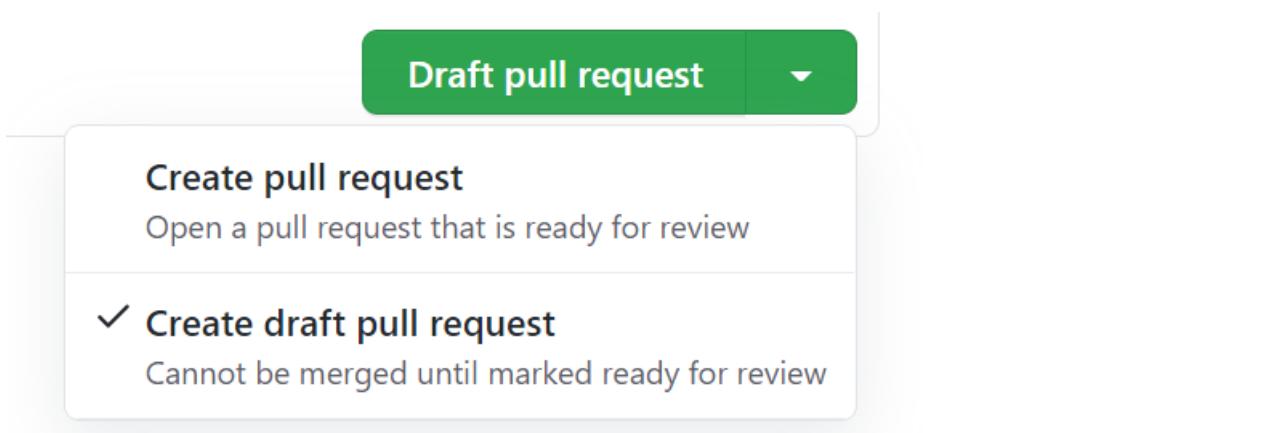
Updates to the table of contents require approval from the repo manager (jimpark).

To add a new node to the Docs contributor guide TOC:

1. Attend the next weekly Contributor guide workgroup meeting to discuss the node. Others in that group may be aware of similar content.  
Contact [Jim Parker](#) for an invite to the next session.
2. Create the pull request in the docs-help-pr repository and include reviewers that you want feedback from.
3. When the reviews are completed and your changes are ready to merge, comment with `#sign-off`. That comment changes the PR label and signals to the PR review team that the PR is ready. The PR review team will do a criteria review and either let you know of issues that need to be addressed, or approve the PR (required) and merge.

## Pull requests

If you expect the work associated with a pull request to last more than a week, create a draft pull request so it's clear that the content is being worked on and not abandoned.



When your additions/updates are ready for review, you must add `#sign-off` per the instructions in the pull request. Using `#sign-off` changes the do-not-merge label to ready-to-merge to let the PR review team know the PR is ready.

### IMPORTANT

PRMerger is turned on for the docs-help-pr repository, but auto-merge is not. All PRs get human review.

## Stale PRs/Issues

- PRs with no activity for 30 days get commented and labeled 'stale-pr'
- PRs that are stale for 10 days are closed
- Issues with no activity for 35 days get commented and labeled 'stale-issue'
- Issues that are stale for 15 days are closed

## Acrolinx in the contributor guide

There's a minimum score of 85 for all articles added or updated in the contributor guide. If you create a pull request that has a score below 85, you need to make the necessary updates to bring the score to 85 or higher.

## Identify a problem

1. Open an [issue](#) in the docs-help-pr repository.
2. Add a title that reflects the topic.
3. In the item description, provide:
  - Your suggestion
  - Other pertinent details

# What's new in the contributor guide

6/9/2021 • 21 minutes to read

Here's a list of recent changes made to the contributor guide.

To find guidance on any other topic, you can also use *Filter by title* present with Search textbox above the Table of Contents(TOC).

If you have questions about, or suggestions for, the guide you can post them in the [Contributor Guides Teams channel](#).

## May 2021

ARTICLE	CHANGE
<a href="#">Respond to inappropriate content</a>	Learn what to do about code of conduct violations and other inappropriate content.
<a href="#">Troubleshooting Git errors</a>	New article with instructions on how to resolve Git-specific errors when updating content.
<a href="#">FAQ best practices</a>	New article providing best practices when writing FAQ content.
<a href="#">Best practices for contributing to docs.microsoft.com</a>	New article outlining important considerations for successfully publishing your content.
<a href="#">Write "best practices" content for using your service</a>	New article providing process guidelines for creating articles about best practices for a service or feature.
<a href="#">SEO onboarding for contributors</a>	New article providing resources to help you incorporate basic SEO into your daily writing practices.
<a href="#">SEO: Checklist for web-friendly docs</a>	New checklist article with guidance for writing web-friendly doc pages.
<a href="#">Get started contributing to Visual Studio documentation</a>	New article providing guidance on how Microsoft employees can contribute to Visual Studio documentation.
<a href="#">Visual Studio docs guidance</a>	New article providing guidance for Microsoft employees who contribute to Visual Studio documentation in visualstudio-docs-pr.
<a href="#">Principles for publishing preview content for Visual Studio and VS Code on docs.microsoft.com</a>	As a rule, Visual Studio and VS Code content does not author preview content on docs.microsoft.com or code.visualstudio.com/docs. This new article provides the reasons why and gives guidance for specific exceptions to this rule.
<a href="#">Banners on docs.microsoft.com</a>	New article about how to request promotional banners to be run on docs.microsoft.com pages.

ARTICLE	CHANGE
<a href="#">Write a Partner Center announcement</a>	New article about how to write announcements that are published to Partner Center.
<a href="#">Getting data on API documentation</a>	Article moved to the docs contributor guide that provides information on how to get data from API documentation content.
<a href="#">Docs.microsoft.com archive, retire, and removal policy</a>	Article moved to the docs contributor guide that provides an overview for the policies on archiving, retiring, and removing articles on docs.microsoft.com.
<a href="#">Information architecture on Docs</a>	Article moved to the docs contributor guide that introduces the core information architecture components used to structure content on docs.microsoft.com.
<a href="#">Archived content</a>	Article moved to the docs contributor guide that provides guidelines and historical context for archived, previous-version content.
<a href="#">Converting SMS content to markdown</a>	Article moved to the docs contributor guide that provides instructions for converting SMS content to markdown.
<a href="#">invalid-value</a>	New article about build validation rule.
<a href="#">languages-missing</a>	New article about build validation rule.
<a href="#">path-achievement-not-found</a>	New article about build validation rule.
<a href="#">pivot-definition-not-found</a>	New article about build validation rule.

## April 2021

ARTICLE	CHANGE
<a href="#">Start contributing to Microsoft Docs</a>	New getting started article with a checklist of items to guide new contributors through required setup, how to edit a doc, and merging changes.
<a href="#">Use style guides for content on docs.microsoft.com</a>	New article providing a list of style guides, and the order you should use them in when writing docs.microsoft.com content.
<a href="#">How to use Microsoft Docs documentation</a>	New article about the docs.microsoft.com platform and the main services of DMC.
<a href="#">Adobe Analytics</a>	New article providing an overview of Adobe Analytics, a comparison with other web analytics options, and support information.
<a href="#">Diagrams for technical communicators</a>	New article providing introduction guidance for creating diagrams.

ARTICLE	CHANGE
How to choose a diagram type	New article about choosing a diagram type, as well as providing diagram instructions and templates to create a diagram.
How to draw a diagram from a diagram type	New article about how to draw a diagram to capture a technical concept.
How to draw a diagram from scratch	New article about how to draw a diagram from scratch, even if you can't draw a realistic illustration freehand.
How to provide feedback to a diagram	New article about how to analyze a diagram, assess what that diagram is communicating, and determine if it is meeting its purpose.
Diagram cookbook	New article providing information for each type of diagram, such as purpose, type, ingredients, and steps.
Diagrams as code	New article providing guidance on how to describe a diagram in text format, such as markdown syntax.
About technical diagram and art	New article about different categories of visualizations that you can use in your documentation.
Process mapping overview	New article providing an overview of how to map a process.
Process definition form	New article providing guidance on using a process definition form.
Process summary format	New article providing a framework for assessing and documenting process effort.
Process map	New article detailing the format for a process map.
docs-link-absolute	New article about build validation rule.
duplicate-h2s	New article about build validation rule.
heading-empty	New article about build validation rule.
heading-no-content	New article about build validation rule.
heading-skipped	New article about build validation rule.

## March 2021

ARTICLE	CHANGE
Add a new article	New article about how to add a new article to docs.microsoft.com.
How to refer to public container images in content	New article providing guidance whenever public container images from Docker Hub are referenced or used in an article.

ARTICLE	CHANGE
Purpose of zero and low page views reports	New article on how to use zero and low page view reports to help you make decisions about whether to retire or archive content.
The Doc Subset dashboard	New article on how to use the Doc Subset dashboard to view KPIs for a content set or set of articles.
Announcements in Partner Center	New article about announcement pages in Partner Center documentation.
Azure Architecture Center refresh	New article providing an overview and guidelines for the Azure Architecture Center content refresh process.
Markdown template for design patterns	New AAC template to create a design pattern.
Markdown templates for reference architectures	New AAC template to create a reference architecture.
Markdown templates for example workloads	New AAC template to create an example workload, example scenario, or sample solution.
Markdown templates for solution ideas	New AAC template to create a solution idea.
Old markdown templates	Updated article to provide examples of deprecated markdown templates from the AAC repo.
dependent-repo-branch-mismatch	New article about build validation rule.
xref-not-found	New article about build validation rule.

## February 2021

ARTICLE	CHANGE
Use multiple redirection files	New article about how to use multiple redirection files in a repo.
How to use the page-level trends & drill-down dashboards	New article about the page-level trends & drill-down dashboards and how to access them.
Write security documentation for an Azure service	New article providing guidelines for writing security documentation for an Azure service doc set.
Base templates for introducing security features supported by an Azure service	Template for introducing key security features supported by an Azure service.
Customer research and experimentation FAQs	New article with answers to frequently asked questions on customer research and experimentation.
include-not-found	New article about build validation rule.
pivot-definition-conflict	New article about build validation rule.

ARTICLE	CHANGE
<a href="#">table-syntax-invalid</a>	New article about build validation rule.

## January 2021

ARTICLE	CHANGE
<a href="#">Write how-to guides</a>	New article providing guidance for writing how-to guides.
<a href="#">Write concepts</a>	New article providing guidance for writing concept articles.
<a href="#">Template for how-to guides</a>	New template for writing how-to guides.
<a href="#">Template for concept articles</a>	New template for writing concept articles.
<a href="#">Contributing to docs.microsoft.com</a>	New article clarifying options for contributing to docs.microsoft.com.
<a href="#">Resolving "stale" GitHub issues</a>	New article that defines the 'stale' issue concept and provides guidance for writers and teams to meet the challenge of managing the backlog of GitHub issues.
<a href="#">Insert and update content from a Jupyter notebook</a>	New article about the Docs Authoring Pack extension for Jupyter Notebooks.
<a href="#">forbidden-content</a>	New article about build validation rule.
<a href="#">forbidden-link</a>	New article about build validation rule.
<a href="#">module-badge-missing</a>	New article about build validation rule.
<a href="#">module-uid-missing</a>	New article about build validation rule.
<a href="#">token-not-found</a>	New article about build validation rule.

## December 2020

ARTICLE	CHANGE
<a href="#">Contributing content to docs.microsoft.com</a>	New article providing an overview and brief description of each high-level phase in the content creation process.
<a href="#">Planning your content</a>	New article listing the tasks of the planning phase of content contribution.
<a href="#">Designing your content</a>	New article listing the tasks of the design phase of content contribution.
<a href="#">Writing your content</a>	New article listing the tasks of the writing phase of content contribution.

ARTICLE	CHANGE
<a href="#">Publishing your content</a>	New article listing the tasks of the publishing phase of content contribution.
<a href="#">Maintaining content</a>	New article listing the tasks of the maintenance phase of content contribution.
<a href="#">Create FAQ content</a>	New article about using the structured YAML template to present FAQ content.
<a href="#">Repository and publishing information for the Partner Center business</a>	New article listing Partner Center repositories.
<a href="#">Azure Architecture Center browser</a>	New article about the Azure Architecture Center (AAC) browser.
<a href="#">New contributor checklist for new content</a>	New article covering the tasks required for publishing a new content set.
<a href="#">Update Azure Databricks content</a>	New article with information about updating Azure Databricks content.
<a href="#">link-redirected</a>	New article about build validation rule.
<a href="#">file-not-found</a>	New article about build validation rule.
<a href="#">sandbox-not-mapped</a>	New article about build validation rule.
<a href="#">redirected-file-not-removed</a>	New article about build validation rule.
<a href="#">quiz-multiple-answers</a>	New article about build validation rule.

## November 2020

ARTICLE	CHANGE
<a href="#">UserTesting</a>	New article about the UserTesting research tool.
<a href="#">Publishing for the air-gapped cloud (AGC)</a>	New article about publishing content in air-gapped locations.
<a href="#">Use applies to and appropriate roles in Partner Center</a>	New article about requirements for Partner Center content.
<a href="#">Connect to CGA Kusto data clusters</a>	New article about requesting access to the Kusto data clusters and connecting to them using client applications.
<a href="#">Community Kusto queries</a>	New article listing community-submitted Kusto queries that other contributors have found useful.
<a href="#">Contributor Insights</a>	New article about the Contributor Insights Dashboard
<a href="#">General tagging guidelines</a>	New conceptual article about metadata tagging guidelines

ARTICLE	CHANGE
<a href="#">module-child-not-found</a>	New article about build validation rule.
<a href="#">path-achievement-not-trophy</a>	New article about build validation rule.
<a href="#">path-child-not-found</a>	New article about build validation rule.
<a href="#">path-child-not-module</a>	New article about build validation rule.
<a href="#">path-modules-missing</a>	New article about build validation rule.
<a href="#">path-trophy-missing</a>	New article about build validation rule.
<a href="#">path-uid-missing</a>	New article about build validation rule.
<a href="#">module-achievement-not-badge</a>	New article about build validation rule.
<a href="#">task-type-invalid</a>	New article about build validation rule.
<a href="#">unit-uid-missing</a>	New article about build validation rule.
<a href="#">duplicate-pivot-groups</a>	New article about build validation rule.
<a href="#">duplicate-pivot-ids</a>	New article about build validation rule.
<a href="#">pivot-group-not-found</a>	New article about build validation rule.
<a href="#">pivot-id-not-found</a>	New article about build validation rule.
<a href="#">pivot-id-unused</a>	New article about build validation rule.
<a href="#">pivot-definition-conflict</a>	New article about build validation rule.

## October 2020

ARTICLE	CHANGE
<a href="#">Edit an article in a private repo</a>	New article about editing content in a private repository using the web browser.
<a href="#">Breadcrumbs guidelines</a>	New conceptual article about breadcrumbs.
<a href="#">Header navigation governance</a>	New conceptual article about header navigation.
<a href="#">Get page-level metrics in your browser: Microsoft Docs Metrics (PREVIEW)</a>	New article about using MDM to get high-level metrics without leaving the page.
<a href="#">What is Content Data Advocacy Group (CDAG)?</a>	New article about the Content Data Advocacy Group.
<a href="#">Taxonomies for Docs</a>	New reference article listing existing metadata taxonomies for Docs.

ARTICLE	CHANGE
<a href="#">List of available contributor guides</a>	New reference article listing known contributor guides.
<a href="#">Error documentation not available</a>	New article about build validation rule.
<a href="#">invalid-tab-group</a>	New article about build validation rule.
<a href="#">redirect-url-invalid</a>	New article about build validation rule.

## September 2020

ARTICLE	CHANGE
<a href="#">sensitive-language</a>	New article about build validation rule.
<a href="#">PR only validation rules</a>	New article about PR only functionality for build validation.
<a href="#">Docs Local Validation</a>	New article explaining how to perform build validations against your local Docs conceptual or Learn repository.
<a href="#">Monitoring content for Azure services</a>	New article about how to write content for monitoring for Azure services.
Base templates for Azure service monitoring article	Template for monitoring for Azure services content.
<a href="#">rule-override-invalid</a>	New article about build validation rule.
<a href="#">Update documentation for a servicing update</a>	New article that explains how to submit a documentation update for a servicing update in the SQL Server repository MicrosoftDocs/sql-docs-pr.

## August 2020

ARTICLE	CHANGE
<a href="#">DocImageSorter</a>	New article about a prototype tool that uses Azure Cognitive Services to identify issues with images.
<a href="#">preserve-view-not-set</a>	New article about build validation rule.

## July 2020

ARTICLE	CHANGE
<a href="#">Unified Content Model FAQ</a>	New article with answers to common questions about the unified content model.
<a href="#">Terminology and inclusive language</a>	Article revised to provide guidelines to address inclusive language in content.

ARTICLE	CHANGE
<a href="#">Major update to video guidance</a>	The video guidance has been brought current and consolidated from 4 articles to 1.

## June 2020

ARTICLE	CHANGE
<a href="#">Is docs.microsoft.com the right fit for you?</a>	Article updated to remove compliance content from the list of content that does not belong in a docs.microsoft.com article or repository.
<a href="#">Planning for and managing costs for an Azure service</a> <a href="#">Azure template for a manage costs conceptual article</a>	New article and template for writing a conceptual article about planning for and managing costs for an Azure service.
<a href="#">Metadata attributes</a>	Article updated with end-user-help added as an ms-topic value.

## May 2020

ARTICLE	CHANGE
<a href="#">Unified content model</a>	A new article about the Unified Content Model.

## April 2020

ARTICLE	CHANGE
<a href="#">Update Azure docs hub page</a>	A new article about how to update the <a href="#">Azure docs hub page</a> .

## January 2020

ARTICLE	CHANGE
<a href="#">Write quickstarts</a>	Article updated and clarified.
<a href="#">Base template for quickstart articles</a>	Template updated and clarified to align with guidance updates.
<a href="#">TwitterApp</a>	New article about using TwitterApp for simplified use and management of public-facing twitter accounts.
<a href="#">Create or update a hub page</a>	Update article to add "sample" to itemType list.
<a href="#">Create or update a landing page</a>	Update article to add "sample" to itemType list.
<a href="#">Metadata attributes</a>	Update article to add 'kb-support' as new allowlist valid value for 'ms.topic'

ARTICLE	CHANGE
<a href="#">Docs Markdown reference</a>	Markdown guidance consolidated.
<a href="#">Meriwether diff reporting tool guide</a>	New article about the Meriwether diff reporting tool. Provides writers the ability to view diff reports for APIs that are new, modified, or deleted for each release.
<a href="#">How to include code in docs</a>	Article updated with new code snippet markdown extension.

## December 2019

ARTICLE	CHANGE
<a href="#">Content and Learning office hours</a>	A new article that lists a current schedule of office hours and guidance for hosting your own office hours.

## November 2019

ARTICLE	CHANGE
<a href="#">Accessibility guidelines for multimedia</a> <a href="#">Conceptual art</a> <a href="#">Docs Markdown reference</a>	Articles cleaned up and updated with new guidance about alt text for icons.
<a href="#">Landing page</a>	Landing page updated to new LP format.
<a href="#">Metadata attributes</a>	"how-to" added as an ms.topic value.

## September 2019

ARTICLE	CHANGE
<a href="#">Revised table of contents</a>	The contributor guide TOC was reorganized to make content easier to find.
<a href="#">Link to articles from the user interface</a>	A new article that describes the standard process to create a contextual help link for use in a user interface, such as the Azure portal.

## August 2019

ARTICLE	CHANGE
<a href="#">Create or update a hub page</a>	A new article about a page that collects related docs.
<a href="#">Create or update a landing page</a>	Update to a page for a single docset.

## May 2019

ARTICLE	CHANGE
<a href="#">Frequently asked questions (FAQ) about reference content</a>	A new article that addresses some common questions about contributing to and understanding reference content on docs.microsoft.com.
<a href="#">How do I contribute to docs.microsoft.com?</a>	A new article for new contributions to docs.microsoft.com.

## April 2019

ARTICLE	CHANGE
<a href="#">Create a landing page</a>	Updates to the Learn module section of this article.

## March 2019

ARTICLE	CHANGE
<a href="#">Markdown cheat sheet</a>	.pdf file removed because the guidance was outdated.
<a href="#">File name and path guidelines</a>	Article updated with content from <i>File names and locations for Azure technical articles</i> which was removed from the guide to eliminate duplication.
<a href="#">Site search enabled for the contributor guide</a>	Scoped search for the docs-help-pr repo enabling search results from the contributor, admin and localization guides when viewing content in the docs-help-pr repo.
<a href="#">Identify preview content</a>	A new article that explains how to identify content for private and public preview.

## February 2019

ARTICLE	CHANGE
<a href="#">Markdown Reference for OPS</a>	Article updated with content from <i>Docs.Microsoft.Com Markdown extension snippets</i> which was removed from the guide to eliminate duplication.
<a href="#">Should I work in a private or a public repository?</a>	Updates to clarify the reasons to work in a private repo.
<a href="#">Request updates to landing pages and hub pages</a>	Updates to clarify existing guidance around landing and hub page requests.
<a href="#">Create a landing page</a>	Updates to clarify existing guidance around landing and hub page requests.
<a href="#">Request approval for an exception or experiment</a>	Outdated guidance removed.
<a href="#">Run Acrolinx locally on Markdown files using the Acrolinx extension for Visual Studio Code</a>	Clean up and consolidate tools articles in the Get started section of the ToC and in Resources/Tools.

ARTICLE	CHANGE
<a href="#">Install content authoring tools</a>	Clean up and consolidate tools articles in the Get started section of the ToC and in Resources/Tools.

## Sprint 144

The cadence for the **What's New** page now follows the regular Content & Learning Sprint dates.

ARTICLE	CHANGE
<a href="#">Tutorials for fork and clone and contributing using VS Code</a>	Major consolidation and updates for setting up your local machine environment and making your first contribution using Visual Studio Code.
<a href="#">Major consolidation to video guidance</a>	The video guidance has been consolidated from 14 articles to 3.
<a href="#">How to plan and host a hack-a-doc</a>	Detailed instructions on planning and hosting a hackadoc event.
<a href="#">Consolidate setup experience</a>	Improve and update instructions for setup and installation of tools.
<a href="#">SEO test runs, validations, and troubleshooting</a>	Instructions for validating and troubleshooting SEO work.
<a href="#">Update Acrolinx score guidelines</a>	Update rules for Acrolinx minimum score by section.
<a href="#">Update installation for new Acrolinx on VSCode</a>	New instructions for installing and using Acrolinx with Visual Studio Code.
<a href="#">Update coding conventions for Azure Powershell</a>	Update and improve the coding conventions for Azure PowerShell.
<a href="#">New .NET Advanced Tutorial template</a>	.NET-specific advanced tutorial document type for more advanced scenarios.

In addition, the external [Contributor Guide](#) now includes the .NET Docs team [guidance for community contributions](#).

## Sprint 143

The cadence for the **What's New** page now follows the regular Content & Learning Sprint dates.

ARTICLE	CHANGE
<a href="#">Localization guidance for legal articles</a>	Update process for localizing legal articles.
<a href="#">How to break up large PRs</a>	New simpler instructions for taking a large PR and breaking it up into smaller, easier to review PRs.
<a href="#">How to request a technical focused review</a>	Formal rollout for the Technical Focused review (TFR) process.

ARTICLE	CHANGE
<a href="#">GitHub issues and PR reports</a>	Instructions on generating and using the GitHub issue and PR aging reports.
<a href="#">New Azure Dev Ops guidance</a>	The repo-specific section for Azure Dev Ops was added.

In addition, the section on using video has been consolidated. The simpler guidance is [Plan a video](#).

We had a number of PRs submitted for the **external contributor guide** from community members. Hacktober may have been a factor. During this sprint, we merged 19 minor PRs, and closed another 9 PRs without merging them.

## August 2018

ARTICLE	CHANGE
<a href="#">When to create a docs troubleshooting article</a>	New guidance for when to create troubleshooting content.
<a href="#">Global template for overview articles</a> <a href="#">Global template for quickstart articles</a> <a href="#">Global template for tutorial articles</a>	New templates added for writing standard content types.
In review status	New review status established for contributor guide articles that propose new processes.   <b>In review</b> This document is not yet a supported part of the content model. We encourage you to test the guidance and provide feedback in <a href="#">Teams</a> .
<a href="#">Contribute to the contributor guide</a>	Updated to include info about adding the "In review" label to identify new processes that need testing and feedback.

## July 2018

ARTICLE	CHANGE
<a href="#">Repo-specific guidance for PowerApps</a>	Removed.
<a href="#">Conceptual art</a>	Updated with information about using the .svg file type.
<a href="#">Include reusable content in articles</a>	New article with best practices for using includes.
<a href="#">Choose the correct content type for your article</a>	New article with explanations of content types (for example, quickstarts, tutorials and how-to articles) and guidance for when to use them.
<a href="#">How to retire, move, or rename a technical article</a>	Updated with information about redirects.
<a href="#">How to redirect obsolete articles</a>	New article about handling redirects.
<a href="#">Accessibility guidelines</a>	Updated with info about the importance of H2s for accessibility and a few guidelines on how to optimize them.

## June 2018

ARTICLE	CHANGE
<a href="#">How to enable the documentation feedback control</a>	Moved article from OPS documentation to the contributor guide.
<a href="#">Contribute to the contributor guide</a>	New article documenting the process for making contributions to the contributor guide.
<a href="#">Writing content for Content &amp; Learning</a>	Updated to emphasize procedures via the CLI for quickstarts, tutorials, and how-to guides in milestone two.
<a href="#">Write a quickstart</a>	Updated to clarify that wait time for things like provisioning don't count against the 10-minute limit.
<a href="#">Accessibility guidelines</a>	New article documenting accessibility guidelines for content contributors.
Alt Text	Moved from the style guide to the contributor guide.
<a href="#">TOC creation and management</a>	New article with details about how to create a TOC.
<a href="#">File name and path guidelines</a>	New article with guidance for file and folder names that are included in the public URL on docs.microsoft.com
<a href="#">The 5 writing principles</a>	Updated with guidance to support adding time estimates in procedures when something will take longer than expected.

## May 2018

ARTICLE	CHANGE
<a href="#">Rename a product or service in content</a>	New article recommends approach for renaming a product or service in technical documentation.
<a href="#">Install content authoring tools</a>	Replaced/updated Gauntlet information with Docs Authoring Pack info.
<a href="#">Request approval for an exception or experiment</a>	New article documenting the process for requesting an exception to the content model criteria, or an experiment to test variations.
<a href="#">Writing principles checklist</a>	New article that provides a list of questions that content developers can ask themselves to assist with aligning their writing to Content & Learning writing principles. Can also be used to make sure content is prepared for a CFR.
<a href="#">How to include code in docs</a>	Article updated to address content overlap. Update to clarify when brackets are used.
<a href="#">Developer quickstart guidance</a>	Articles updated to address content overlap.
<a href="#">Add in-line code snippets</a>	Deleted as part of the code content overlap resolution.

## ARTICLE

## CHANGE

<a href="#">Subscriptions - Azure subscriptions for writers</a>	Updated Azure internal registration and subscription (AIRS) guidance and added more vendor info.
---	--

## April 2018

ARTICLE	CHANGE
<a href="#">Contributor's guide</a>	Updated TOC and landing page.
<a href="#">Create a lightbox</a>	Added new article on using Lightbox to include expandable images in content.
<a href="#">PowerApps docs</a>	Added article covering repo-specific information for PowerApps.
<a href="#">Power BI docs</a>	Added article covering repo-specific information for Power BI.
<a href="#">Flow docs</a>	Added article covering repo-specific information for Flow.
<a href="#">Contribute to SQL docs</a>	Added new article covering repo-specific information for SQL.
<a href="#">SQL versioning guidance</a>	Added new article covering repo-specific information for SQL.
<a href="#">TOC Guidance for Docs Content</a>	Updated with additional content type definitions.
<a href="#">C + AI technical content types</a>	Article removed from the contributor guide - use <a href="#">TOC Guidance for Docs Content</a> instead.
<a href="#">Conceptual art</a>	Consolidated articles covering art topics to remove outdated and duplicate information.
<a href="#">Screenshots: How to create, format, and embed in documentation</a>	Added GIMP and Paint.Net tools and instructions. Added best practice item for hiding internal version of Azure portal when capturing screen images.
<a href="#">Create and publish animated GIFs in documentation</a>	Added new article on animated GIFs.
<a href="#">Technical content channel guidance</a>	Added links to appropriate blog guidance.
<a href="#">Hub and Landing Page Guidance for content</a>	Deleted and redirected to <a href="#">Create a landing page</a> .
<a href="#">Request updates to landing pages and hub pages</a>	Updated to align with current guidance and with onboarding guide <i>Plan your hub/landing page</i> article.
<a href="#">How to include code in docs</a>	New article for people new to putting code in docs. Covers in-line snippets, best practices for things like code line lengths, etc.

ARTICLE	CHANGE
<a href="#">Write samples</a>	Revised to clarify the process for creating articles in the Samples node of the TOC.

## March 2018

ARTICLE	CHANGE
<a href="#">Use release branches</a>	Updated with guidance for using release branches
<a href="#">Write an overview</a>	Updated with guidance to include the customer intent statement as a comment in the metadata section.
<a href="#">Write a quickstart</a>	Updated with guidance to include the customer intent statement as a comment in the metadata section. Clarified that quickstarts are not required if no meaningful functionality can be introduced to new users in under 10 minutes. Clarified distinctions between quickstarts and how-to articles.
<a href="#">Write a tutorial</a>	Updated with guidance to include the customer intent statement as a comment in the metadata section. Clarified distinctions between tutorials and how-to articles.
<a href="#">TOC Guidance for Docs Content</a>	Updated the definition of quickstarts and how-to guides.
<a href="#">Create a screenshot</a>	Updated with guidance for using animated GIFs.
<a href="#">Create and publish animated GIFs</a>	Added new article about using animated GIFs in documentation
<a href="#">Text formatting guidelines</a>	Added new article about formatting text (when to use bold, italics, code, etc.)
<a href="#">GitHub for content reviews</a>	Updated with guidance for responding to PR comments - add a thumbs up emoji if you agree with a comment, to avoid sending an unnecessary email.

## February 2018

ARTICLE	CHANGE
<a href="#">White papers</a>	Added new article about hosting white papers.
<a href="#">Set up a local Git repository</a>	Updated guidance.
<a href="#">Create a landing page</a>	Updated to align with existing guidance.
<a href="#">Write a quickstart</a>	Revised article to be less Azure specific.
<a href="#">Write a tutorial</a>	Revised article to be less Azure specific.

ARTICLE	CHANGE
<a href="#">Request a CFR</a>	Updated with guidance for creating a customer intent statement.
<a href="#">Sovereign clouds</a>	Added new article about sovereign clouds.
<a href="#">Style and grammar</a>	Added new article about style and grammar.
<a href="#">Link blog to article</a>	Added new article about linking blogs to articles.

## January 2018

ARTICLE	CHANGE
<a href="#">Install content authoring tools</a>	Updated guidance for installing Git and VS Code.
<a href="#">Create a landing page</a>	Updated with guidance about limiting the number of quickstarts listed on landing page to 5 Updated with guidance about limiting the number of tutorials listed on landing page to 10

## 2017

ARTICLE	CHANGE
<a href="#">Write an overview</a>	Updated with guidance about adding "overview" to Page title, Meta description, H1, Intro paragraph
<a href="#">Write a quickstart</a>	Updated with guidance about adding "quickstart" to Page title, Meta description, H1, Intro paragraph
<a href="#">Write a tutorial</a>	Updated with guidance about adding "tutorial" to Page title, Meta description, H1, Intro paragraph

# Contributing to docs.microsoft.com (DMC)

3/22/2021 • 3 minutes to read

This article helps you choose the best path for contributing to content in [docs.microsoft.com](#) (DMC).

MicrosoftDocs content is hosted and managed through a publishing process via GitHub, and there are a couple of ways to contribute depending on the type of change you want to make.

## NOTE

To publish a new content set to MicrosoftDocs, review the steps in the [Admin guide](#) and [complete this form](#).

## Minor updates

If you want to make small content changes, are new to content contribution, or contribute occasionally, you can directly [edit the article in a web browser](#). This workflow is used for one time fixes such as misspelled words or other small changes. All you need is a GitHub account. Since the edits are made directly on the webpage, it's simple and quick to use.

The typical process for a minor update is illustrated below:



## Major updates

The major update workflow is used when you add new content or make significant changes. You can also add images and tables to the markdown source files.

The typical process for a major update is illustrated below:



### One-time setup

There are a few things you need to do to make major updates, but most of these steps you only need to do once:

1. [Create a GitHub account](#)
2. [Install authoring tools](#)
3. [Prepare your environment](#)

These steps provide you with the appropriate permissions and tools to make significant changes to your content.

### Recurring steps

These things you'll do *every time* you want to update:

- [Make changes](#)

- Create a pull request

## Private or public repository

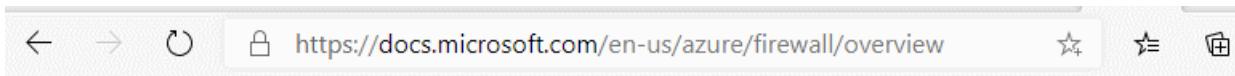
If you're a content developer or work directly for the product team whose content you want to update, you should work in the private GitHub repository, if one exists. A private repository enables your content changes to be staged for preview, SME review, link validation, Acrolinx review, and faster publishing. And small changes in the private repository can be merged automatically.

Contributing to the public repository requires more time and manual resources to complete the publish process. However, if you don't work closely with the product team, you should make your contributions in the public repository. Doing so ensures your changes will get appropriate review and support.

### Finding private repos

Many (not all) public repos for docs have paired private repos that provide a richer contributor experience for Microsoft employees, including preview builds, automated grammar and style checking (Acrolinx), flexibility to collaborate privately inside the company, and use of automation such as `#sign-off` commenting to merge pull requests. For more about the differences between public and private repos for *Microsoft Docs*, see [Should I work in a private or a public repository?](#)

To get to the private repo for a page, add *review* to the beginning of the URL on the published page (the Microsoft Docs page, not the GitHub page), and press **Enter**. Selecting the *Edit* link now sends you directly to the private (-*pr*) repo, if one exists, for the documentation.



You can also append *-pr* to the repo name in the GitHub URL. Note, however, the *-pr* naming convention is standard in *most* cases; not all Microsoft Docs repos follow this convention.

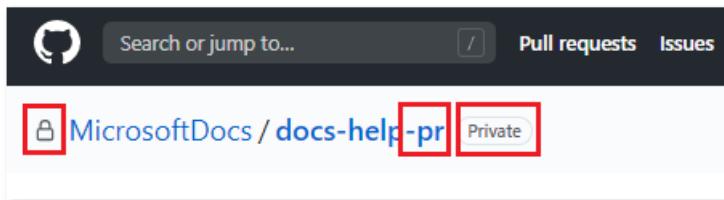
#### TIP

For a step-by-step guide on editing in **public** repos, see [Edit an article on Docs](#).

### Example: Public vs private repo URLs

SOURCE	URL
Published Docs	<a href="https://docs.microsoft.com/azure/firewall/overview">https://docs.microsoft.com/azure/firewall/overview</a>
Published Docs (linking to - <i>pr</i> repo)	<a href="https://review.docs.microsoft.com/azure/firewall/overview">https://review.docs.microsoft.com/azure/firewall/overview</a>
Public GitHub repo	<a href="https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/firewall/overview.md">https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/firewall/overview.md</a>
Private GitHub repo	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/firewall/overview.md">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/firewall/overview.md</a>

To indicate the private status of the repo, the repo name (usually) has a `-pr` suffix, and on the GitHub repo page there's a **Private** label after the name and a lock icon before the name.



## Next steps

[Setup GitHub account](#)

# Start contributing to Microsoft Docs

5/17/2021 • 3 minutes to read

If you're not a full-time writer, and you want to contribute to Microsoft Docs, this article is for you. This checklist guides you through required setup, how to edit a doc, and then add (merge) your changes to our published documentation. There are two paths you can follow, depending on the extent of the changes you want to make:

- Minor changes are edits to a single document. To make a minor change:
  1. Start with the [GitHub setup](#)
  2. [Edit content](#) using the GitHub UI
  3. Publish your changes by making a [Pull request](#)
- Major changes include adding files or images, removing files, or editing more than one document. To make major changes:
  1. Start with the [GitHub setup](#)
  2. [Author or edit content in VS Code](#)
  3. Publish your changes by making a [Pull request](#) that goes through validation and review steps.

## GitHub setup

You only need to do this once. Complete the following step:

	TASKS
<input checked="" type="checkbox"/>	<a href="#">Set up a GitHub account</a>

## Edit content

### Minor changes: Use the GitHub UI

If you have minor edits to a single document, working in the GitHub WebUI is the recommended method.

	TASKS
<input type="checkbox"/>	<a href="#">Make quick edits in a private Microsoft Docs repository using the GitHub WebUI.</a>
<input type="checkbox"/>	As a Microsoft employee, make sure that you're editing on the private GitHub repository. GitHub private repositories are internal versions of our docs, and provide build validation and docs preview prior to publish. To help ensure this, install the <a href="#">SpineEdit browser extension</a> . For more information, see <a href="#">SpineEdit</a> .

### Major changes: Use Visual Studio Code

This setup allows you to make large content changes using Visual Studio Code (VS Code). You don't need to do these steps if you're using the GitHub WebUI.

	TASKS	FREQUENCY
<input type="checkbox"/>	<a href="#">Install content authoring tools</a>	One time setup.

	TASKS	FREQUENCY
<input type="checkbox"/>	<a href="#">Fork and clone the repository</a>	Required for each repository you want to edit.
<input type="checkbox"/>	<a href="#">Edit an article</a>	Required for each repository you want to edit.
<input type="checkbox"/>	<a href="#">Author an article</a>	Required for each repository you want to add content to.

You're now set up to edit an article using [Visual Studio Code](#).

#### TIP

Use the [Git command cheat sheet](#) for help with command line tips.

### Learn the basic writing guidelines

- As an internal Microsoft contributor, make changes on the [private repo](#). Don't make changes on the public repo.
- Make changes on a [working branch](#). Don't make changes on default or main branches.
- Use [correct text formatting](#).
- Use correct [image naming and formatting](#).
- [Link correctly](#).
- Use [everyday words](#).
- Refer to the [style guidelines](#) for a list of style guides and the order to use them in.
- If you add a new file, update the table of contents.
- If you delete a file, [add a redirect](#).

### Publish your changes

After you make changes in an article, create a pull request to merge these changes to the main branch. After the pull request is created, the PR goes through [validation](#) and review before being merged. For more information, see [What happens when I submit a PR?](#)

Each repository has its own best practices for the review flow of new PRs. Contact the [article owner](#) or find other relevant contributors using the [GitHub contributor alias search](#).

	TASKS
<input type="checkbox"/>	<a href="#">Create and edit pull request</a>
<input type="checkbox"/>	Address any validation errors. For more information, see <a href="#">Docs Build validation overview</a> .
<input type="checkbox"/>	Add reviewers by @mention their alias in a comment. Use the <a href="#">GitHub contributor alias search</a> .
<input type="checkbox"/>	Interact with <a href="#">Pull request comments and suggestions</a>
<input type="checkbox"/>	<a href="#">Review and sign off</a>

The vendor PR reviewer team is available during the [review and publishing schedules](#).

For more information, you can take a look at the full [Docs contributor guide](#).

## Other: Repo-specific guidance

Some repositories have their own specific processes and guidance. Refer to the [Repo-specific guidance](#) section in the Docs contributor guide and determine whether you are publishing content to any of those repos. Not all content have repo-specific guidance, but if yours does, make sure to follow the guidance. Some common repo-specific processes you'll find in this section include:

- Joining security groups
- Getting access to specific repos
- Tracking your content work in Azure DevOps

Most contributions will go in an existing repository, but there may be a need to create a new repo. For more information on creating a new repository, see the [Repo Admin Guide](#).

## Next steps

- [Quick reference for contributing to the docs](#)
- [Learning path - contribute to Microsoft Docs](#)
- Still have questions? You can find a list of further resources on the [Resources for help](#) page.

# Set up a GitHub account

3/27/2021 • 4 minutes to read

To contribute to [MicrosoftDocs](#) content you'll need to create your own GitHub account and link it to your Microsoft account. You'll join the organization that holds the source files of the article(s) you want to update. Also, you'll complete security protocols including two-factor authentication and create access tokens in addition to configuring email notifications. **You'll only need to do this once.**

## TIP

Refer external contributors (Non MSFT) to [this public contributor guide page](#) for similar information. Please help us keep the information in sync between these two pages as things change.

## Create a GitHub account and profile

Go to [github.com/join](https://github.com/join) and create your account. If you already have a GitHub account, skip to [link your accounts section](#).

Create your profile as follows:

- **Profile picture:** A picture that represents you.
- **Name:** Your first and last name.
- **Email:** Your Microsoft email address. If you already have an account associated with a private email address, link to your Microsoft account.
- **Company:** Microsoft Corporation

## Link your GitHub and Microsoft accounts

Link your GitHub to your Microsoft account to contribute to [docs.microsoft.com](https://docs.microsoft.com).

1. Sign in to your [GitHub account](#).
2. Check your [Microsoft Open Source linking](#) status:
  - If your accounts are already linked, you see **You're linked!** and you can skip to the next section.
  - If not, follow the onscreen prompts to link your GitHub account to your Microsoft account.
  - After you link your GitHub account with your Microsoft account, you'll authorize your [Personal Access Tokens](#) and [authorize SSH keys for corporate repos](#).

## Join MicrosoftDocs and other repositories

While anyone with a GitHub account can contribute to Docs content using a public repository, including external customers, Microsoft employees should use private repositories when available. [Private or a public repository](#) can help you decide which repository to use.

To contribute to [docs.microsoft.com](https://docs.microsoft.com), join the MicrosoftDocs organization.

1. Go to [Microsoft Open Source organizations](#).
2. Ensure you are in **Organizations** tab, scroll down to the **Available organizations** section.
3. Locate **MicrosoftDocs** and then select **Join**. The Want to join MicrosoftDocs page appears, select **Join**.
4. Repeat for all other organizations you want to collaborate on, such as **Microsoft** or **Azure**.

5. If the default "everyone" team assignment for the organization you joined is sufficient, skip to the next section.
6. If your work group requires that you join a **specific team**:
  - a. In the Open Source organizations page, select the **Teams** tab.
  - b. Scroll to locate your team, or use the search field, enter the team name given to you by your GitHub administrator and search.
  - c. Locate the team in the search results list, select the name and then click the **Request to join** button.  
For every organization you chose in step 3 and 4, you'll also receive an email invite from GitHub. You can also click the **Join-OrganizationName** button in the email to join.

## Enable two-factor authentication

You must enable two-factor authentication (2FA) on your GitHub account to contribute to private/internal doc repos.

Follow these instructions to set up 2FA: [Securing your account with two-factor authentication \(2FA\)](#) GitHub article. As a convenient cheatsheet, these are the quick steps to follow (always refer to preceding link as official guidance):

1. Read and follow [Configuring 2FA via a TOTP mobile app](#).
2. [Download & save your 2FA recovery codes](#) if you didn't already. **Do NOT lose these recovery codes!**
3. [Set a fallback SMS number for authentication](#), in case you lose access to your primary device and recovery codes.

The first time you authenticate to GitHub from Git, you'll be asked to [provide your 2FA authentication code](#) using the TOTP mobile app you configured in this procedure.

## Authorize the Docs Portal to access your GitHub account

Docs (formerly known as OPS) is our documentation platform that builds content from documentation repositories. The platform translates Markdown and YAML into HTML, and publishes it to [docs.microsoft.com](https://docs.microsoft.com) and [review.docs.microsoft.com](https://review.docs.microsoft.com) (the staging/preview site).

Docs provides extensions to standard Markdown syntax to provide special features such as alert boxes and tabs. Following is an alert box.

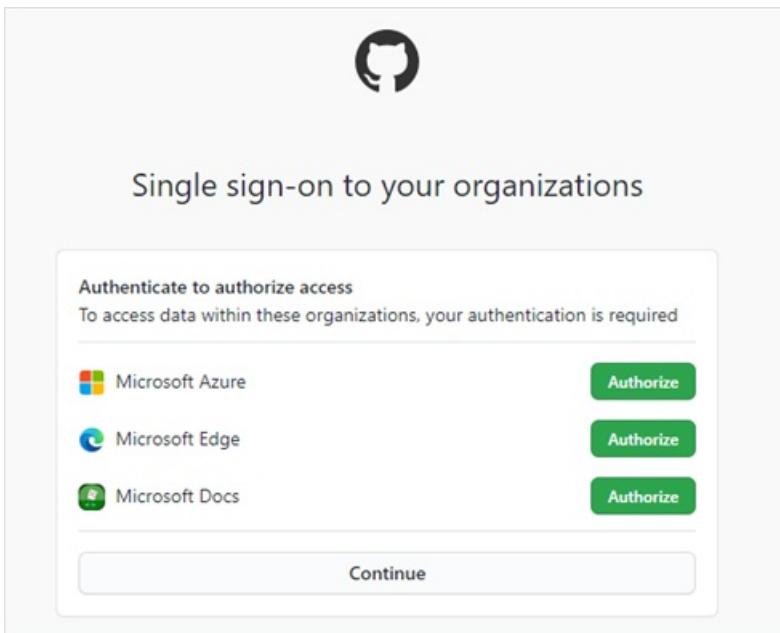
```
>[ !WARNING]  
> Here's an alert box.
```

### WARNING

Here's an alert box.

You must authorize the Docs build system to access your GitHub account profile information. The authorization allows the build system to access profile details such as your Microsoft email address and repositories permissions.

1. Go to the [Docs Portal](#).
2. Select the **Sign in with GitHub** option.
3. Complete the 2FA authentication. The single sign-in to your organization page appears with a list of organizations you selected in the Join MicrosoftDocs and other repositories section.



4. Select **Authorize**. When you're authorized, you'll be directed to the [Docs Portal](#) home page. Close the page.

## Configure notifications from Docs Portal

To receive email notifications from the Docs Portal build service, you need to modify some of your personal account settings on GitHub.

In GitHub, select the profile drop-down in the upper right and choose **Settings**.

### To get email for push operations

1. Go to the **Emails** tab in the left nav.
2. Clear the **Keep my email address private** check box.

When this option is unchecked and you push commits, GitHub sends Docs Portal a notification with the primary email address in this list. Otherwise, GitHub sends Docs Portal a notification with `username@users.noreply.github.com`, and you won't receive a notification from Docs Portal.

#### NOTE

In the [Install content authoring tools](#) article you will add the check mark again.

### To get email for pull requests

1. Select the **Profile** in the left nav.
2. Set a **Public email**.

For a pull request, GitHub doesn't send Docs Portal the user's email address. So, Docs Portal locates the username to get the user's public email address, which is listed in this field. If you don't list one, you won't get mail for your pull request.

### To not receive additional notifications

- Select **Emails** in the left nav.
- In the **Email preferences** section, select **Only receive account related emails, and those I subscribe to**.

## Next steps

Make minor updates Install authoring tools for major updates

# Edit an article in a private repo

5/6/2021 • 9 minutes to read

You can submit quick edits for review by making a pull request. Once a pull request is opened, you can discuss potential changes with collaborators and the content authors. In this article, you'll learn how to edit in **private** repos by submitting edits via a pull request. A pull request (PR) is a request for a content owner to *pull* into the official source contributions you have made to your copy of a project (your *fork*).

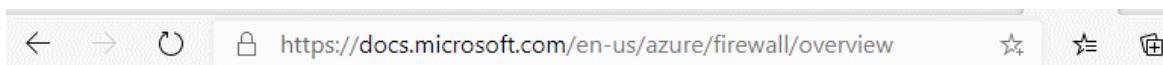
## Prepare your GitHub environment

You need a [GitHub](#) account to contribute to Microsoft Docs.

- If you're an internal contributor, clearly identify yourself as a Microsoft employee when you set up your account.
  - Include a picture of yourself and your first and last name
  - Use your Microsoft email-if you already have an account and it's associated with a private email, add your Microsoft email as a secondary address.
  - For **Company**, enter *Microsoft Corporation*.
- To contribute to docs.microsoft.com, link your Microsoft account with your GitHub account
  - Using the Open Source portal, [link your GitHub to your Microsoft account](#).
  - After your accounts are linked, [join the MicrosoftDocs organization](#).
- To contribute to private/internal docs, enable two-factor authentication (2FA)
  - To securely verify your identity, download the [Microsoft Authenticator app](#)
  - On GitHub, [enable two-factor authentication](#)
- Finally, authorize the OPS build system to access your GitHub account
  - Authenticate with your GitHub credentials on the [OPS portal](#)
  - Verify that **MicrosoftDocs** is listed under **Organization access**. Select the **Authorize openpublish** button.

## Workflow for editing a private repo

1. Sign in to GitHub.
2. Go to the page of the Microsoft Docs article you want to update. To work on the private repo version of the article, add "*review:*" to the beginning of the URL. For more information on private repos, see [Finding private repos](#), below.



3. Select **Edit** (top-right) to go to a GitHub page with the source of the article.

## NOTE

### 404 errors

To access a private repo, you must be [signed in to GitHub](#) and you must have access permissions for that repo. If you're signed in and you're getting a 404 error when attempting to access a private repo, this either means the repo doesn't exist, or it does, and you don't have access. To get access, your GitHub account needs to be configured with two-factor authentication *and* joined to the correct organization (such as [MicrosoftDocs](#)). For more information, see [GitHub account setup](#).

4. Select **Edit this file** (the pencil icon) on the page of the GitHub source file. If you're a user with write access, you can contribute directly to the repo. If you don't have write access, a new forked repo is created in your GitHub account—if you don't already have one—and you'll see a message reminding you of the new fork.

You're making changes in a project you don't have write access to. Submitting a change will write it to a new branch in your fork `rey-u/docs-help-pr`, so you can send a pull request.

5. Edit the file. Microsoft Docs articles are written in Markdown (`.md`); here is the [Docs Markdown reference](#). Select **Preview changes** to ensure your content is rendering as expected.
6. Scroll to the bottom of the page, where you see one of the following:
  - **Propose file change:** Appears when you have read-only access to the repository, such as [editing files in another user's repository](#). In this case, GitHub will create a "patch" branch in your fork of the repository (and automatically create a fork if it doesn't exist). After you select **Propose file change**, a **Comparing changes** page appears so you can verify your changes. Then select the **Create pull request** button to submit your changes to the pull request queue.
  - **Commit changes:** Appears when you have write permissions to a repository, such as [editing files in your own repository](#). In this case, GitHub gives you two options for saving your changes:
    - **Commit directly to the `<branch-name>` branch**, where `<branch-name>` is the name of the branch that you were on when you selected the **Edit** button. This applies your changes directly to the branch instead of using a pull request. (At this point, you're finished!)
    - **Create a new branch for this commit and start a pull request**, which prompts you with a default name to create a new branch. After you select **Propose file change**, a **Comparing changes** page appears so you can verify your changes. Then select the **Create pull request** button to submit your changes to the pull request queue.

## Pull request processing

The previous section walked you through the process of submitting proposed changes, by bundling them in a new pull request (PR) that is added to the destination repository's PR queue. A pull request enables GitHub's collaboration model, by asking for the changes from your working branch to be pulled and merged into another branch. In most cases, that other branch is the default/master branch in the main repository.

### Validation

Before your pull request can be merged into its destination branch, it might be required to pass through one or more PR validation processes. Validation processes can vary depending on the scope of proposed changes and the rules of the destination repository. After your pull request is submitted, you can expect one or more of the following to happen:

- **Mergeability:** A baseline GitHub mergeability test occurs first, to verify whether the proposed changes in your branch are in conflict with the destination branch. If the pull request indicates that this test failed, you must reconcile the content that is causing the merge conflict before processing can continue.
- **CLA:** If you are contributing to a public repository and are not a Microsoft employee, depending on the magnitude of the proposed changes, you might be asked to complete a short Contribution License Agreement (CLA) the first time you submit a pull request to that repository. After the CLA step is cleared, your pull request is processed.
- **Labeling:** Labels are automatically applied to your pull request, to indicate the state of your pull request as it passes through the validation workflow. For instance, new pull requests might automatically receive the "do-not-merge" label, indicating that the pull request has not yet completed the validation, review, and sign-off steps.
- **Validation and build:** Automated checks verify whether your changes pass validation tests. The validation tests might yield warnings or errors, requiring you to make changes to one or more files in your pull request before it can be merged. The validation test results are added as a comment in your pull request for your review, and they might be sent to you in e-mail.
- **Staging:** The article pages affected by your changes are automatically deployed to a staging environment for review upon successful validation and build. Preview URLs appear in a PR comment.
- **Auto-merge:** The pull request might be automatically merged, if it passes validation testing and certain criteria. In this case, you don't need to take any further action.

### Review and sign-off

After all PR processing is completed, you should review the results (PR comments, preview URLs, etc.) to determine if additional changes to its files are required before you sign off for merging. If a PR reviewer has reviewed your pull request, they can also provide feedback via comments if there are outstanding issues/questions to be resolved prior to merge.

By assigning the appropriate label to a pull request, comment automation lets read-level users (users who don't have write permissions in a repo) do write-level actions. If you're working in a repo that has comment automation turned on, you can use hashtag comments to assign labels, change labels, or close a pull request. Comment automation will also notify Microsoft employees by email for review and sign-off of public repository PRs, whenever someone proposes changes to articles where you're the author.

HASHTAG COMMENT	WHAT IT DOES	REPO AVAILABILITY
#sign-off	When the author of an article types <b>#sign-off</b> in the comment stream, the <b>ready-to-merge</b> label is assigned. This label lets the reviewers in the repo know when a pull request is ready for review/merge.	Public and private
#sign-off	If a contributor who <i>isn't</i> the listed author tries to sign off on a pull request in a public repo, comment automation writes a message to the pull request indicating that only the author can assign the label.	Public
#hold-off	If authors change their mind or make a mistake, they can type <b>#hold-off</b> in a PR comment to remove the <b>ready-to-merge</b> label. In the private repo, <b>#hold-off</b> assigns the <b>do-not-merge</b> label.	Public and private

HASHTAG COMMENT	WHAT IT DOES	REPO AVAILABILITY
#please-close	Authors can type #please-close in the comment stream to close the pull request if they decide not to have the changes merged.	Public
#label:"custom label text"	Authors can add a custom label up to 200 characters (shorter recommended).	Public and private

When the pull request is issue-free and signed off, your changes are merged back into the parent branch and the pull request is closed.

## Publishing

Remember, your pull request has to be merged by a PR reviewer before the changes can be included in the next scheduled publishing run. Pull requests are normally reviewed/merged in the order of submission. If your pull request requires merging for a specific publishing run, you will need to work with your PR reviewer ahead of time to ensure that merging happens prior to publishing.

After your contributions are approved and merged, the docs.microsoft.com publishing process picks them up. Depending on the team that manages the repository you are contributing to, publishing times can vary. Articles published under the following paths are normally deployed at approximately 10:30 AM and 3:30 PM Pacific Time, Monday-Friday:

- <https://docs.microsoft.com/azure/>
- <https://docs.microsoft.com/aspnet/>
- <https://docs.microsoft.com/dotnet/>
- <https://docs.microsoft.com/enterprise-mobility-security>

It can take up to 45 minutes for articles to appear online after publishing. After your article is published, you can verify your changes at the appropriate URL:

`http://docs.microsoft.com/<path-to-your-article-without-the-md-extension> .`

## Troubleshooting

If your fork is not up-to-date with the upstream, you might receive one of the below errors when you edit article and submit a PR:

- 500 error
- *This blob took too long to generate* error
- Merge conflicts

To avoid and resolve errors, keep your fork up-to-date with the upstream.

### Steps to sync your fork with upstream branch from the GitHub web UI

1. Open your fork of the repository.
2. Select **Fetch upstream > Fetch and merge** button

The screenshot shows a GitHub fork page for the repository `nivnar/docs-help-pr-1`. The page indicates that the branch is 148 commits behind `MicrosoftDocs:master`. A list of commits is shown, all made by `Jim-Parker`, including updates to `.github`, `archive`, `help-content`, `tools`, and Acrolinx configuration files. The repository has 105 branches and 0 tags. On the right side, there are sections for **About** (Documentation and guidance for onboarding and contributing to docs.microsoft.com), **Releases** (No releases published, Create a new release), **Packages** (No packages published, Publish your first package), and **Languages** (PowerShell 79.6%, Python 14.7%, C# 5.7%).

If there are no merge conflicts between the branches, your fork's branch is updated by merging from the upstream's branch. If there are conflicts, you will be prompted to open a pull request to resolve.

## Finding private repos

Many (not all) public repos for docs have paired private repos that provide a richer contributor experience for Microsoft employees, including preview builds, automated grammar and style checking (Acrolinx), flexibility to collaborate privately inside the company, and use of automation such as `#sign-off` commenting to merge pull requests. For more about the differences between public and private repos for *Microsoft Docs*, see [Should I work in a private or a public repository?](#)

To get to the private repo for a page, add `review.` to the beginning of the URL on the published page (the Microsoft Docs page, not the GitHub page), and press **Enter**. Selecting the *Edit* link now sends you directly to the private (`-pr`) repo, if one exists, for the documentation.

You can also append `-pr` to the repo name in the GitHub URL. **Note, however,** the `-pr` naming convention is standard in *most* cases; not all Microsoft Docs repos follow this convention.

### TIP

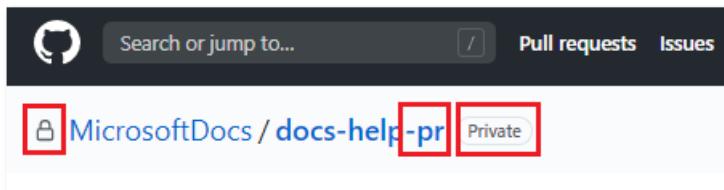
For a step-by-step guide on editing in public repos, see [Edit an article on Docs](#).

### Example: Public vs private repo URLs

SOURCE	URL
Published Docs	<a href="https://docs.microsoft.com/azure/firewall/overview">https://docs.microsoft.com/azure/firewall/overview</a>
Published Docs (linking to <code>-pr</code> repo)	<a href="https://review.docs.microsoft.com/azure/firewall/overview">https://review.docs.microsoft.com/azure/firewall/overview</a>
Public GitHub repo	<a href="https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/firewall/overview.md">https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/firewall/overview.md</a>

SOURCE	URL
Private GitHub repo	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/firewall/overview.md">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/firewall/overview.md</a>

To indicate the private status of the repo, the repo name (usually) has a `-pr` suffix, and on the GitHub repo page there's a **Private** label after the name and a lock icon before the name.



# Install content authoring tools

3/22/2021 • 4 minutes to read

This article lists and describes how to install the recommended tools you'll need to efficiently make in-depth and large content changes. For simple and one-time changes to articles, [edit in your browser](#) without installing these tools.

## TIP

For external contributors refer to [this public contributor guide page](#) for similar information. Please help us keep the information in sync between these two pages as things change.

## Prerequisites

- [Set up a GitHub account](#)

## Install Git client tools for Windows and macOS

There are essentials tools for both Windows and macOS, that you should install.

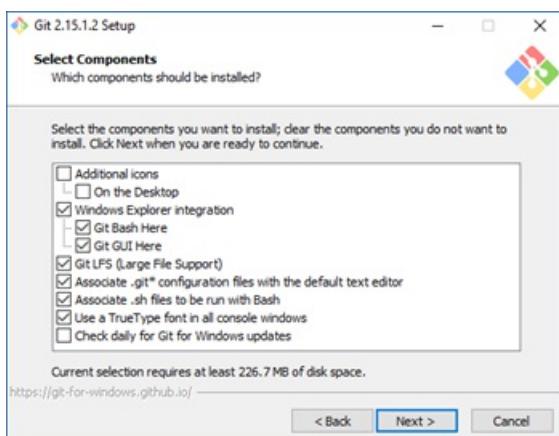
### Install Git client tools for Windows

#### NOTE

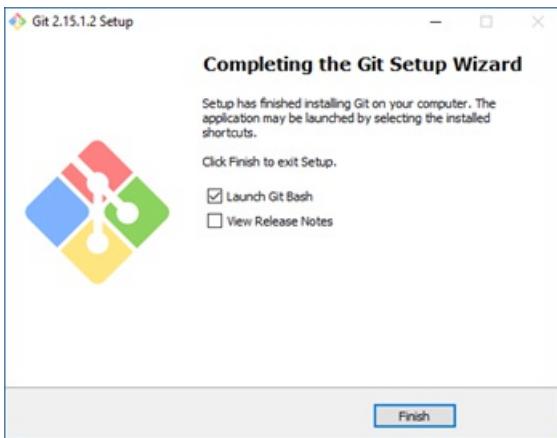
- If you prefer a graphical user interface (GUI) over a command-line interface (CLI), [Visual Studio Code](#), [Software Freedom Conservancy's available GUI Clients page](#), and [GitHub's GitHub Desktop](#) offer that functionality.

1. Download [Git for Windows](#).

2. Run the downloaded executable (.EXE) file and follow the prompts to install. Select **Next** at each prompt to accept all default settings.



3. Select **Finish** to complete the installation.



After installing the Git for Windows, you'll have to configure your Git name and your email address, before installing Visual Studio Code.

### Install Git client tools for macOS

[Homebrew](#) is a package manager that is available on macOS.

1. Install Homebrew using the following command-line script:

```
/usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

#### WARNING

Running arbitrary script from the internet isn't generally a safe practice, always ensure that the script is safe to run and that it is using a secure protocol (such as [https](https://))

2. Run the following command to install Git:

```
brew install git
```

After installing and you'll have to configure your Git name and your email address.

## Configure your Git name and email address

#### NOTE

If you are an existing user, before you update to the *noreply email address*, check that you have pushed all local commits on all branches on all repos. Failure to do so may result in error message '*remote: error: GH007: Your push would publish a private email address*' when pushing existing commits containing your original email address. To remediate the issue, temporarily clear the **Block command line pushes that expose my email** setting, push the changes, and then select it again.

In a browser, go to GitHub.com and follow these steps to configure your Git name and email address.

1. Sign in to [GitHub](#) using your GitHub account.
2. From your **Profile** drop-down select **Settings**.
3. Select **Emails** from the left nav. Your email settings appear. Shortcut: <https://github.com/settings/emails>.
4. Make sure this option is checked: **Keep my email addresses private**
5. Make a note of your *noreply* email address (`<UID>+<github_name>@users.noreply.github.com`).

Keep my email addresses private 

We'll remove your public profile email and use `<noreply_email_address>@users.noreply.github.com` when performing web-based Git operations (e.g. edits and merges) and sending email on your behalf. If you want command line Git operations to use your private email you must [set your email in Git](#).

- Now go to Git bash to configure your Git details using your noreply email address.

```
git config --global user.name "My Name"  
git config --global user.email "<noreply_email_address>"
```

- List your local settings to ensure the user.name and user.email values are correct.

```
git config --list
```

## Install Visual Studio Code on Windows

[Visual Studio Code](#) is a lightweight editor that works on Windows, Linux, and Mac.

### NOTE

To install Visual Studio Code on macOS, we recommend that you use Homebrew.

- Download and install Visual Studio Code:
  - [Windows](#)
  - [Linux](#)

## Install Visual Studio Code on macOS

To install Visual Studio Code using Homebrew, you need to use the `brew cask install` command as it is distributed in a different manner than Git.

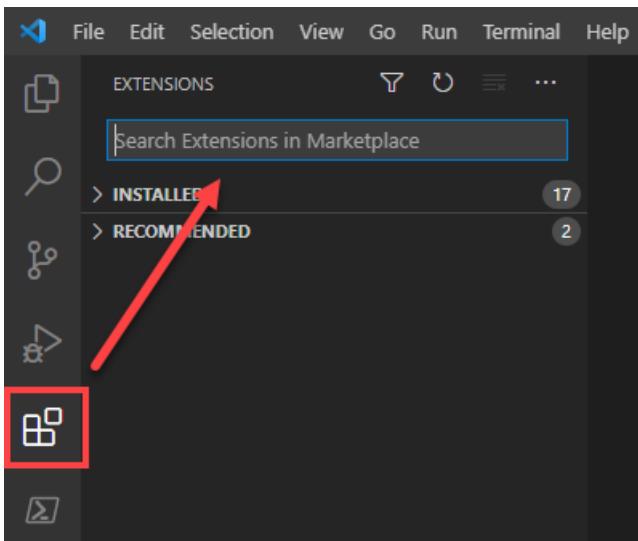
- Run the following command to install Visual Studio Code:

```
brew cask install visual-studio-code
```

## Install Visual Studio Code extensions

To install the extensions:

- Start Visual Studio Code.
- Select the square **Extensions** icon on the left nav. The **Extensions: Marketplace** pane appears.
- In the **Search Extensions in the Marketplace** search box, type the name of an extension you want to find.



4. In the results that appear, locate the extension you want and select **Install**.

### Install Docs Authoring Pack

The Docs Authoring Pack for Visual Studio Code includes basic Markdown authoring assistance, page previews, support for Markdown templates, markdownlint, and Code Spell Checker.

To install the Docs Authoring Pack, click **Install** from the [Docs Authoring Pack page](#) in the VS Code Marketplace.

To use the Docs Authoring Pack functionality, press Alt+M in Visual Studio Code. To configure a toolbar to show the functions available, edit the Visual Studio Code settings (Control+comma), and add user setting

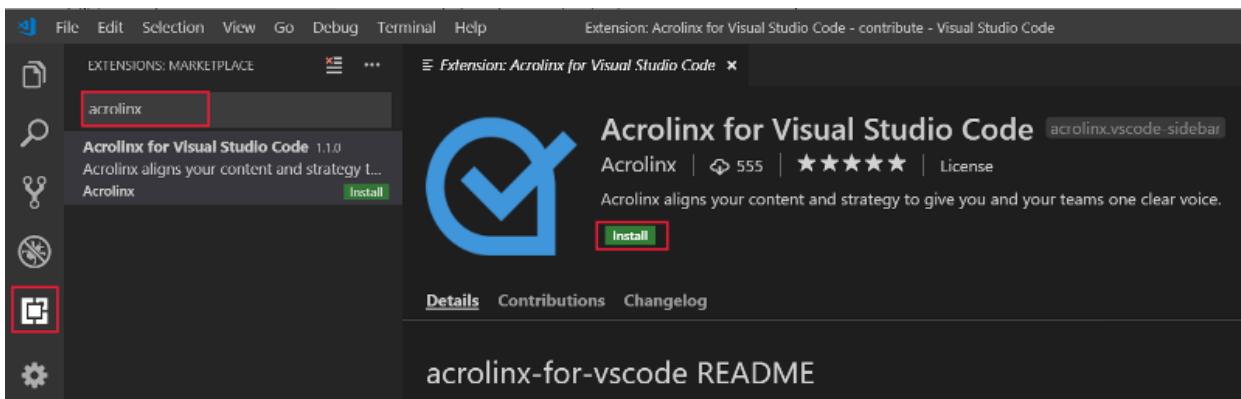
```
"markdown.showToolbar": true
```

For more information, see [Docs Authoring Pack for Visual Studio Code](#).

### Install Acrolinx

The Acrolinx extension for Visual Studio Code checks spelling, grammar, style, tone, clarity, and key terminology usage.

Acrolinx is also available from [Visual Studio Marketplace](#).



When you install Acrolinx in Visual Studio Code, you'll need to configure the server details the first time you run it. For information about installing the Acrolinx extension, and configuring the server details, see [Run Acrolinx locally on Markdown files using the Acrolinx extension within Visual Studio Code](#).

### Install posh-git (optional)

Posh-git is a PowerShell module that integrates Git and PowerShell by providing Git status summary information that can be displayed in the PowerShell prompt.

**On Windows:** [Install posh-git](#) and [review further reading](#). You can run the install command from within Visual Studio Code PowerShell view. In Visual Studio Code, select **View** and then choose **Terminal**. The PowerShell window opens in the bottom of the screen.

On macOS: Install the posh-git equivalent, [bash-git](#).

## Next steps

[Prepare your environment](#)

# Preparing your environment

3/22/2021 • 5 minutes to read

This article shows how to set up your local repository environment so that you can contribute to MicrosoftDocs. There are three steps: *fork* the source files from a repository (repo). You'll then *clone* your fork and set up the *remotes*. This process is for large changes; for simple edits and changes, you can [edit docs in your web browser](#).

## Fork → Clone → Remotes

### TIP

Refer external contributors to [this public contributor guide page](#) for similar information. Keep these two pages in sync as things change.

### Prerequisites

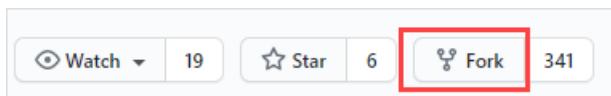
- Identify the GitHub repository that stores your team's documentation
- [To set up a GitHub account](#)
- [To install authoring tools](#)

Here's a video demonstration of the fork, clone and the remote setup process described below:

## Fork the repository

To *fork* a repository means to create a copy of it in your *GitHub* account. You'll have write access to the fork in your account. The fork enables you to make pull requests from your fork to the production repository.

1. Sign in to [GitHub](#).
2. Go to the GitHub repository page (for example <https://github.com/microsoftdocs/docs-help-pr>). If you get a 404 error on the repository page URL, make sure that you're signed in to GitHub. If you're signed in, verify that you have the necessary [permissions](#) to access the repository.
3. From the upper right corner, select **Fork**. A dialog box appears.



4. In the response to **Where should we fork**, select your account.

### Rename your fork

You might have to fork multiple repositories with the same name that live in different orgs. GitHub will by default give a unique name to the new fork if you're forking a second repository with the same name. If you don't like the default name, you can rename it in GitHub.

1. On your fork's home page, select **Settings**.

A screenshot of a GitHub repository settings page. The URL is `/docs-help-pr`. The page shows tabs for Code, Pull requests, Actions, Projects, Wiki, Insights, Settings (which is highlighted with a red box), and Security.

2. Enter a new Repository name, and then select Rename.

A screenshot of the GitHub settings page under 'Repository name'. The input field contains 'ak-setup-edits' with a green checkmark icon next to it. A 'Rename' button is visible to the right.

## Clone the repository

*Clone* means create a copy of a repository on *your* local machine.

1. Create a GitHub folder in your local machine. For example `c:\GitHub`
2. In a browser, go to [GitHub.com](#) and sign in.
3. In GitHub on the top right, from your profile drop-down, select **Your repositories**. The repositories page appears with a list of your forked repositories.
4. Select the repository you want to clone. The repository page of *Your fork* appears.

Make sure you're on the repository page for *your fork*, **not the page for the production repo**. You can confirm that the repository you're on is your fork by checking the URL: it should show your GitHub user name, not the Microsoft organization. For example, <https://github.com/nancydavolio/azure-docs-pr> and **NOT** <https://github.com/MicrosoftDocs/azure-docs-pr>.

A screenshot of a GitHub repository page for a fork. The URL is `Your name / docs-help-pr`. The page shows tabs for Code, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. A red arrow points to the 'Code' tab. A red box highlights the copy icon next to the HTTPS URL in the code editor interface.

5. Select the `<> Code` tab at the top left of the page.
6. On the right-side, from the **Code** drop-down, copy the URL under the **HTTPS** option.

A screenshot of the GitHub 'Code' dropdown menu. The 'Clone' option is selected. Below it, the HTTPS URL `https://github.com/Your name / docs-help-pr` is listed with a copy icon to its right, which is highlighted with a red box.

### IMPORTANT

You copied the clone URL via the GitHub UI. The rest of the steps should be performed in **VS Code Terminal**.

### To clone from Visual Studio Code

1. Open Visual Studio Code (VS Code).
2. From top nav menu, select **View** and then select **Terminal**. The terminal view appears in the bottom of

the VS Code screen.

3. In the Terminal view, run the command to change directory (cd) to the GitHub folder you created for repositories on your drive in Step 1 in the Clone a repository section.

Example:

```
cd c:\GitHub
```

If you get an error, you may have to change directory one folder at a time.

```
PS C:\Users\ashwi> cd Documents  
PS C:\Users\ashwi\Documents> cd GitHub
```

4. Run the following command to clone the repository:

```
git clone <paste the clone URL you copied via the GitHub UI>
```

For example:

```
git clone https://github.com/nancydavolio/azure-docs-pr.git
```

5. When prompted to authenticate via the browser, select **Yes** and complete the process. The clone command downloads the repository files from your fork into a new folder in the folder location you chose in Step 3.

#### NOTE

If you plan to run Acrolinx in Visual Studio Code, the path to the clone must include the repository name. This normally happens by default. When you clone name-docs-pr, the repository goes into a folder named name-docs-pr. For repositories that have non-unique names, your local clone must be in a path that includes the GitHub organization. For example, the ASP.NET and .NET documentation repositories must be in a location that includes aspnet/docs or dotnet/docs in the local file path.

### Authentication errors

If you're cloning a private repository, you may see a message similar to the following example:

```
Cloning into 'docs-help-pr'...  
remote: The `nancydavolio` organization has enabled or enforced SAML SSO. To access  
remote: this repository, visit https://github.com/enterprises/microsoftopensource/sso?  
authorization_request=PMNZGKZDFNZ2GSYML52HS4DFVNHW5  
remote: and try your request again.  
fatal: unable to access 'https://github.com/tdykstra/docs-help-pr.git/': The requested URL returned error:  
403
```

Open a browser, go to the URL provided, and sign in. Then return to your VS Code Terminal and run the `clone` command again.

## Set up remotes

*Remotes* are aliases that refer to remote repositories. You'll use *origin* to refer to your fork, and *upstream* to refer to the production repository. Git automatically created the `origin` remote when you ran the `clone`

command. Run the following commands in either **Git Bash** or **VS Code Terminal**.

1. Change directory (`cd`) into the repository folder that the `clone` command created. For example:

```
cd repo-name-pr
```

2. Run the following command to add a remote named `upstream` that points to the production repository.

For example, the `upstream` repository URL will be <https://github.com/MicrosoftDocs/azure-docs-pr> rather than <https://github.com/nancydavolio/azure-docs-pr>.

```
git remote add upstream <the clone URL of the main repo>
```

For example:

```
git remote add upstream https://github.com/MicrosoftDocs/azure-docs-pr.git
```

3. Run the following command to make sure your remotes are correct:

```
git remote -v
```

Example of output:

```
origin  https://github.com/nancydavolio/azure-docs-pr.git (fetch)
origin  https://github.com/nancydavolio/azure-docs-pr.git (push)
upstream    https://github.com/MicrosoftDocs/azure-docs-pr.git (fetch)
upstream    https://github.com/MicrosoftDocs/azure-docs-pr.git (push)
```

4. If you made a mistake, remove the remote and repeat the step to add it. To remove `upstream`, run the command:

```
git remote remove upstream
```

## HTTP 500 errors

If you get HTTP 500 errors while using your fork, try deleting and re-creating the fork in private mode when you use the Edge or Chrome browsers. To use private mode in the Chrome browser, select the vertical ellipsis and then **New incognito window**. In the Edge browser, select the ellipsis and then **New InPrivate window**. To delete a fork, use the same procedure as for any repository: go to the fork's GitHub page and select **Settings** > **Delete this repository**.

## Next steps

[Add new article](#) [Make changes to an article](#)

# Add a new article

5/7/2021 • 3 minutes to read

This article describes how to add a new article using Visual Studio Code (VS Code) and how to create a new article using a content template.

## Prerequisites

- Identify the GitHub repository that stores your team's documentation.
- [Setup your GitHub account](#).
- [Install authoring tools](#).
- [Prepare your environment](#).

## Create and check out your working branch

Whenever you add a new article, ensure you're **not** on the main or default branch. Remember to create a working branch.

Creating a *Fork*, *Clone* and a *Remote* are typically a one-time set up, however, you would either need a new *Working branch* or would refresh your working branch every time you make content updates. It's always a good practice to create your own *working branch* to make your changes. We don't recommend making changes to the master/main branch directly.

1. In VS Code, open the repository folder of your local clone. From the **File** menu, select **Open folder** and navigate to the folder.
2. Select **View** from the top menu and then select **Terminal** to open the integrated terminal.
3. In the integrated terminal, make sure you're in the repository folder.

### TIP

When you create a working branch from a remote, you will do a pull request for that remote branch and not the default/master branch.

1. To switch to the master branch in your local repository, run the following command:

```
git checkout master
```

2. To ensure your local master branch is current with everything in the production repository's master branch, run the following command:

```
git pull upstream master
```

3. To create a local working branch based on master, run the following command:

```
git checkout -b nameofbranch
```

4. To create the working branch in your GitHub fork, run the following command. The **-u** option links the local and remote branches. Then, when you have this branch checked out, you can push commits to your fork by entering just `git push` instead of `git push origin nameofbranch`.

```
git push origin nameofbranch -u
```

## Add a new article from scratch

Follow the instructions below to create a new article in VS Code. These instructions will create an empty Markdown file in which you can add your content. If you'd like to start a new article from a template go to the next section.

To add a new article through VS Code:

1. Open VS Code.
2. From the **View** menu, select **Terminal**. The PowerShell command prompt appears in the bottom of the screen.
3. Ensure you are in your working branch, see the bottom left for the branch indicator.



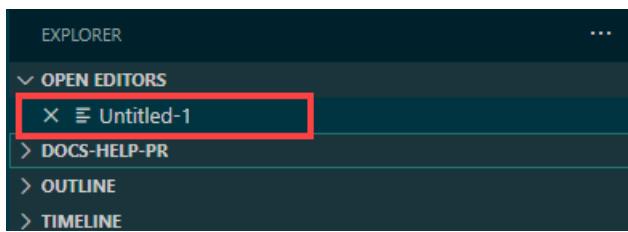
4. If you need to switch to a working branch, run the following command within the VS Code Terminal View:

```
git checkout branch name
```

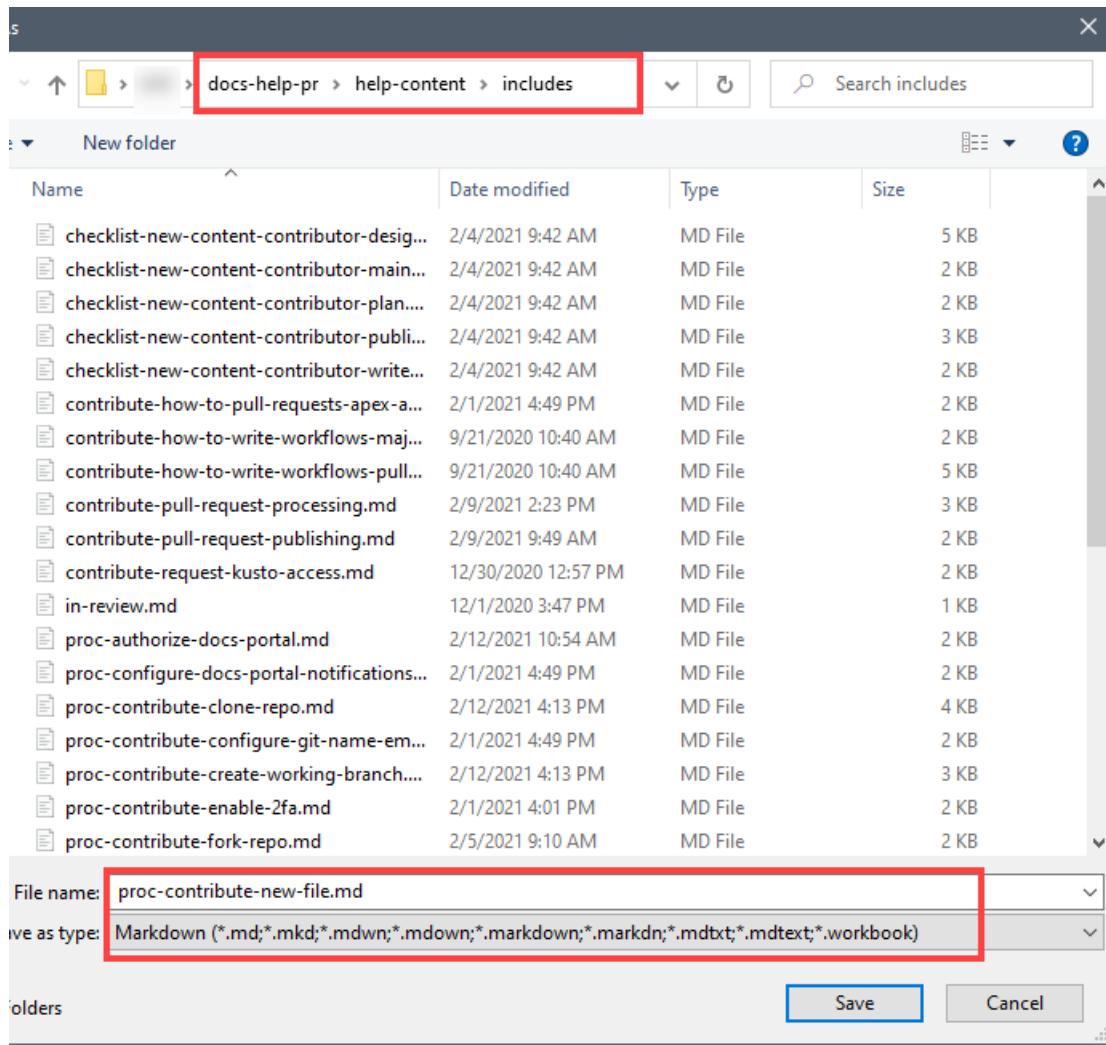
5. Ensure your working branch is current with the master branch, run the following command within the VS Code Terminal View:

```
git pull upstream master  
git push origin master
```

6. To add a new file, select the **File** menu and then choose **New file**. A new untitled file appears in the Explorer view on the left under Open Editors.



7. To name the file, select **File** and then choose **Save**. The Save As dialog box appears. Navigate to the right location for this file before saving.
8. VS Code defaults to .txt file type, change to .md for markdown. Your new file is now created.

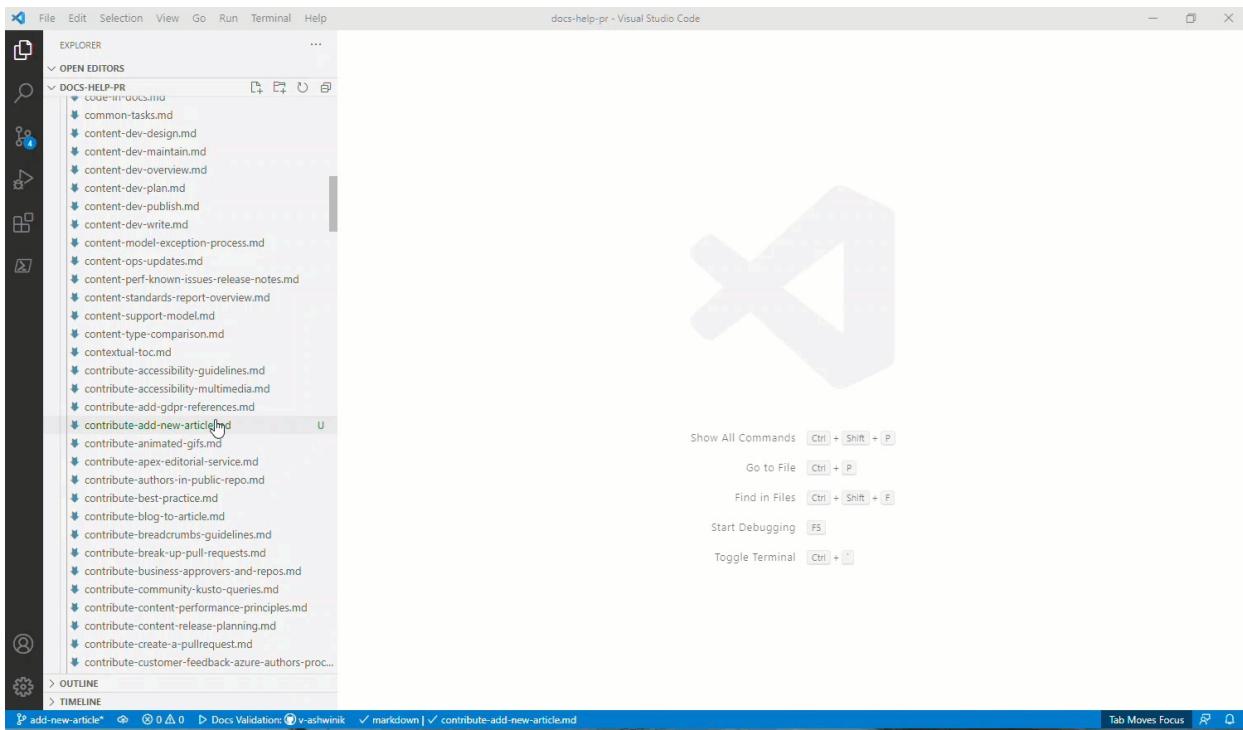


9. In your new file, before you begin your content remember to add the [metadata values](#). Open any markdown file and copy the metadata to your file. Make appropriate changes.

10. Add your content in the markdown file and save.

## Create a new article using a content template

Follow the instructions below to create a new article using one of the content templates in Markdown code. The template provides guidance to address the key elements of the article such as, Metadata tags, headlines, and the overall article flow.



1. From VS Code, open any Markdown file.
2. Press the **Alt & M** keys to open the Docs Markdown menu. The menu appears at the top of the VS Code screen.
3. Select **Template** from the menu. A list of article templates appears.
4. Select the article template you want, such as **Docs template: How-to article**. A new Markdown file is created, populated with the template text.
5. Select **File** choose **Save As**, add File name. Select **Save**. Save the file first to use Docs Markdown features such as inserting relative links.
6. Edit the file, using the template as a guide.

## Next steps

Commit and push your changes

# Edit an article

6/15/2021 • 9 minutes to read

This article shows you how to change a docs.microsoft.com article using Visual Studio Code. You'll edit the Markdown source files and run Git commands. For an abridged "cheat sheet" of these commands, see [Basic Git command line steps](#).

## Prerequisites

- Identify the GitHub repository that stores your team's documentation.
- [Setup your GitHub account](#).
- [Install authoring tools](#).
- [Prepare your environment](#).

## Create and check out your working branch

Whenever you edit a file, ensure you're **not** on the main or default branch. Remember to create a working branch.

Creating a *Fork*, *Clone* and a *Remote* are typically a one-time set up, however, you would either need a new *Working branch* or would refresh your working branch every time you make content updates. It's always a good practice to create your own *working branch* to make your changes. We don't recommend making changes to the master/main branch directly.

1. In VS Code, open the repository folder of your local clone. From the **File** menu, select **Open folder** and navigate to the folder.
2. Select **View** from the top menu and then select **Terminal** to open the integrated terminal.
3. In the integrated terminal, make sure you're in the repository folder.

### TIP

When you create a working branch from a remote, you will do a pull request for that remote branch and not the default/master branch.

1. To switch to the master branch in your local repository, run the following command:

```
git checkout master
```

2. To ensure your local master branch is current with everything in the production repository's master branch, run the following command:

```
git pull upstream master
```

3. To create a local working branch based on master, run the following command:

```
git checkout -b nameofbranch
```

4. To create the working branch in your GitHub fork, run the following command. The **-u** option links the local and remote branches. Then, when you have this branch checked out, you can push commits to your

fork by entering just `git push` instead of `git push origin nameofbranch`.

```
git push origin nameofbranch -u
```

## Locate the source Markdown file

There are multiple ways to locate the source markdown file:

**Option 1** - To edit an article, find the source file for the article you're going to work with in your local repository clone. Within VS Code, the content (.md/Markdown) files from the repository are accessible. Locate the file and open to edit.

**Option 2** - If you're unable to locate the file, visit the article in <https://docs.microsoft.com> and select the Edit link. The relative folder location in the GitHub repository shows in the URL. Here's an example Edit link URL:

```
https://github.com/Microsoft/azure-docs/blob/master/articles/azure-functions/functions-overview.md
```

Here's an example file location for this URL. The repository name is different because the Edit link goes to the public repository.

```
C:\GitHub\*\azure-docs-pr\articles\azure-functions\functions-overview.md
```

**Option 3** - If there is no edit option on an article, append the URL with `review`. to access the private repo version of the article.



**Option 4** - You can also locate the source file through the meta tag `original_content_git_url` in the article source page. With the article open in a browser, right-click on the browser and select **View page source** (ctrl+U). In the source page, search for the meta tag `original_content_git_url`. Here's an example of the meta tag:

```
<meta name="original_content_git_url" content="https://github.com/MicrosoftDocs/docs-help-pr/blob/master/help-content/contribute/markdown-reference.md" />
```

## Edit the file

1. Once you've identified the file name from the URL, locate it within VS Code.
2. Open the file in VS Code, and make your changes.
3. Save your changes. (By default, VS Code doesn't save changed files on exit.)

## Run Acrolinx

Acrolinx is a service that checks an article for grammar and style issues. It makes suggestions for improvements and calculates a number grade. In some repositories, the grade must be 80 or above. The Acrolinx extension is a required tool as mentioned in the *Install authoring tools* article, which is a [prerequisite for this article](#). Verify with your team that Acrolinx use is permitted in your team's repository.

1. Select the **Acrolinx** icon on the left sidebar of the Visual Studio Code screen.



2. In the Acrolinx window, select the **Check** tab and then the **Check** button.
3. Review the issues and make more changes if you want to address any of the issues.

## Preview the article

It's often helpful to get an advance look at how the Markdown will be rendered on the docs.microsoft.com site.

1. On the top right of the article page within VS Code, click the **Open preview** icon. The preview tab appears next to the article tab.
2. If you have the VS Code legacy toolbar enabled, click the preview icon on the bottom of the VS Code UI. The preview tab appears side-by-side to the article.
3. Alternatively, press **Ctrl+Shift+V** keys to open the preview tab.
4. To view the preview from the command palette, press **Ctrl+Shift+P** to open the Command palette.
5. Enter **Docs:** in the search box.
6. Select **Docs: Preview**. The preview tab appears next to the article tab.
7. Verify that your change looks as expected.

### TIP

- If you prefer, you can use the keyboard shortcut Alt+DP (hold down Alt and press D, then release Alt and press P), or the **Preview** icon in the Docs Authoring Pack menu bar.
- There is also a built-in **Markdown preview**, but the **Docs preview** is better. The Docs preview expands code snippet references and applies docs.microsoft.com site style to alerts, such as this **Tip** box. The Markdown preview is what you get when you select the **Open Preview to the Side** icon at the top right of the screen or by pressing **Ctrl+KV**.

## Commit the change

1. Run the `git status` command to verify that, only the files you edited appear in the list of changed files.

```
git status
```

2. Once you have verified the files, run the `git add` command followed by the *file path* and *file name* to stage the files you've changed.

```
git add folder-name/file-name.md
```

These commands:

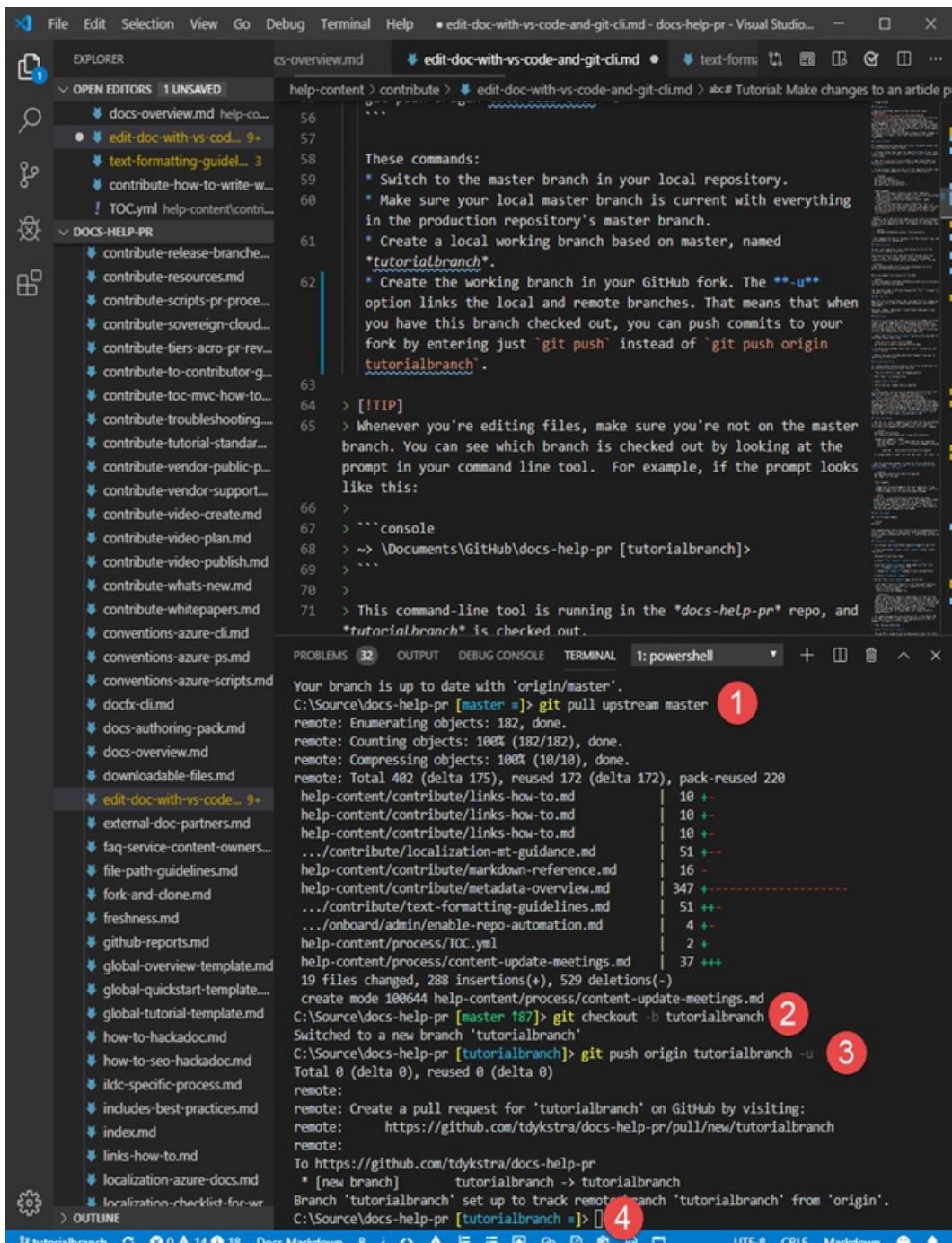
- Stage the file you changed in the local repository. If you changed multiple files, enter a `git add` command for each file.
  - Create a commit with the specified commit message.

## **NOTE**

`git add .` (note the period after `add`) is an alternative to specifying file names manually. It automatically stages all of the changes you've made in the repo. It's quicker to use but can cause problems by including changes that were made by accident. If you choose to use `git add .`, remember to run `git status` beforehand so that you know what changes will be staged.

## PoshGit to review Git status

The following image shows the commands entered in the Visual Studio Code integrated terminal. The green and blue text in brackets is how posh-git keeps you informed of Git status:



1. At the start, posh-git shows that the main/default branch is checked out and it's in sync with the writer's fork (origin).
2. After `git pull upstream`, posh-git shows that main/default branch is still checked out and the local clone is 87 commits ahead of main/default branch in origin. That means there are now 87 commits in the local clone that haven't been pushed to main/default branch in origin. If you want to keep your local main/default branch in sync with upstream, you can run `git push`.
3. After `git checkout -b tutorialbranch`, posh-git shows that tutorialbranch is checked out.
4. After `git push origin tutorialbranch -u`, posh-git shows that tutorialbranch in the local clone is in sync with tutorialbranch in origin.

The following example shows how posh-git indicates file changes:

```
C:\Source\AspNetCore.Docs [signalrts3 ≡]>
C:\Source\AspNetCore.Docs [signalrts3 ≡ +0 ~3 -0 !]> git add .
C:\Source\AspNetCore.Docs [signalrts3 ≡ +0 ~3 -0 ~]> git commit -m "fee 2 ck
[signalrts3 fcbb6b69e] feedback
 3 files changed, 5 insertions(+), 5 deletions(-)
C:\Source\AspNetCore.Docs [signalrts3 ↑1]> git push 3
Enumerating objects: 21, done.
Counting objects: 100% (21/21), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 825 bytes | 275.00 KiB/s, done.
Total 11 (delta 8), reused 0 (delta 0)
remote: Resolving deltas: 100% (8/8), completed with 8 local objects.
To https://github.com/tdykstra/AspNetCore.Docs
 766a9336e..fcbb6b69e  signalrts3 -> signalrts3
C:\Source\AspNetCore.Docs [signalrts3 ≡]> 4
```

1. After three files are edited, posh-git shows `~3` in red, meaning the changes are not yet staged.
2. After the files are staged, the `~3` is in green.
3. After the staged changes are committed, the local branch is one commit ahead of its remote tracking branch.
4. After the commit is pushed to the remote branch, local and remote branches are in sync again.

#### TIP

Whenever you're editing files, make sure you're not on the main/default branch. As the preceding image shows, posh-git makes it easy to see which branch is checked out and whether it's in sync with its remote tracking branch. This automatic reminder of your current Git status is one reason why posh-git is recommended for command-line Git use.

## Troubleshooting

If you get HTTP 500 errors while using your fork, try deleting and re-creating the fork in private mode for Edge or Chrome. To use private mode in Chrome, select the vertical ellipsis and then New incognito window. In Edge, select the ellipsis and then New InPrivate window. To delete a fork, use the same procedure as for any repository: go to the fork's GitHub page and select Settings > Delete this repository.

## Push the commit

Run the following command:

```
git push
```

This command pushes the current branch to the corresponding branch in the `origin` remote. It knows which

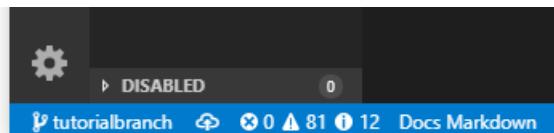
remote and which remote branch to push to because you used the `-u` option when you did the first push.

After you have pushed the changes, you'll need to sign in to GitHub.com, go to your repository, and [create a pull request](#).

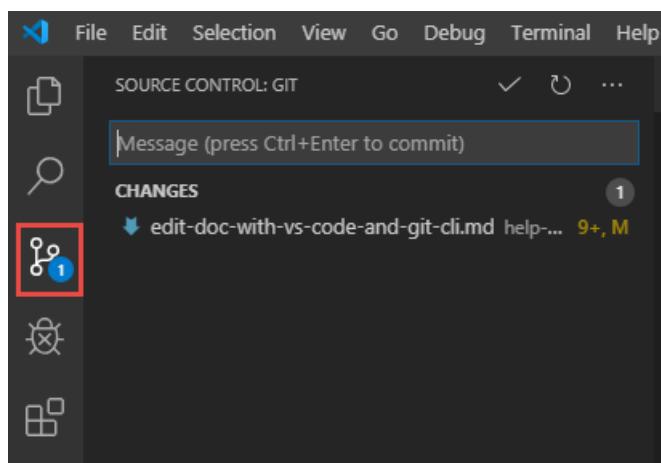
### Use the VS Code UI to push and commit

You can also create and push a commit by using Git integration features of Visual Studio Code. Remember to continue your work on the same machine. If you switch machines, you won't have the latest version of the working branch to continue with.

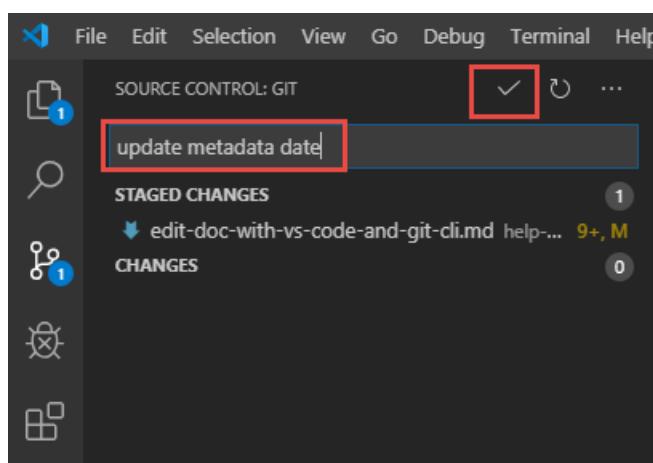
1. Make sure that your local repo still has the same working branch checked out. In the Visual Studio Code UI, the branch is shown at the lower left. The branch name also functions as a button for switching branches.



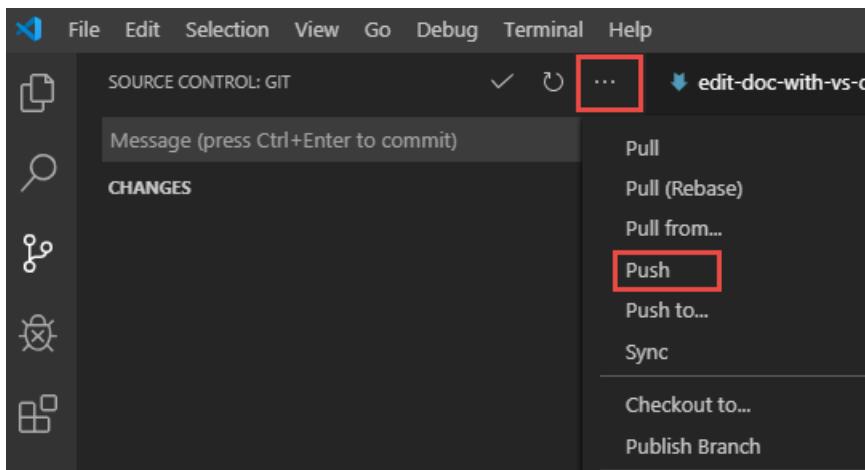
2. Open the same file in the editor again, and make the second planned change to the doc.
3. If you made substantial edits or did a freshness review, update the date in the metadata block at the top of the file. Be sure to keep leading zeroes in the month and day fields (mm/dd/yyyy format).



4. In the VS Code left side bar, select the fork tool to see the Source Control: Git sidebar.
5. The file you edited appears under **Changes** with an **M** on the right to indicate it is modified. New files would have **U** (untracked) and deleted files would have **D**.
6. Stage the changes. Hover over the **Changes** line, and you see a **+** icon. Select that icon. This is equivalent to the `git add` command. The file now appears in the **Staged Changes** section.



7. Enter a commit message such as "update metadata date" in the **Message** text box, and select the check mark (or press Ctrl+Enter).



8. Push the commit. Select the ellipsis at the right of the **Source control: Git** heading, and then select **Push**.
9. Look at the pull request page in GitHub, and you see that there are now two commits. Because you added the commit to the branch specified in the PR, GitHub automatically incorporates it into the PR. In the publishing system, the change triggers a build, and the PR changes are validated and staged again.
10. On the GitHub pull request page, select the **Files** tab. The *diff* display shows both of your changes.

## Next Steps

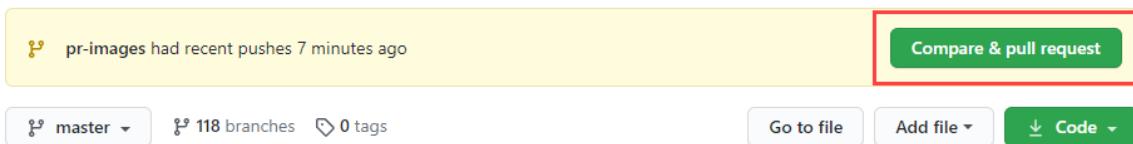
[Create your pull request](#)

# Create a pull request

6/9/2021 • 9 minutes to read

You create a pull request (PR) after you've updated or added your content. This step is required for getting your changes published. You'll push the changes from Visual Studio Code and continue to GitHub to complete the creation of the pull request.

1. In a browser, go to the GitHub.
2. Select your profile on the top right and select **Your repositories**.
3. If there are more than one, select the repository for your fork. For example, `yourgithubid/docs-help-pr`.
4. If you see a green **Compare & pull request** button, select that button.



Otherwise follow these steps:

- a. Select **Pull requests > New pull request**.
- b. On the **Comparing changes** page, make sure the **head fork**: drop-down is set to your fork.
- c. Change the **compare**: drop-down to your working branch.
- d. Select **Create pull request**.

## NOTE

Alternatively, if you are making changes in a release branch, the *base*: branch should be changed to your release branch. Your changes will be merged to the base branch.

5. On the **Open a pull request** page, verify that:

- The *base*: branch is *master/main/default* in the *upstream* repository. All your changes will be merged to the upstream branch.
- The number of commits and files changed is what you expect.

## IMPORTANT

When creating pull requests, always make sure that only the commits and files you expect to be included are included. Also make sure that the PR goes to the correct branch in the upstream repository. Most of the time, *master/main/default* is the right branch, but sometimes you'll be working with a release branch. A release branch groups many changes intended to be released simultaneously. On the release date, the release branch is merged to *main/default*.

## Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

The screenshot shows the GitHub pull request creation interface. At the top, there are dropdown menus for 'base repository' (MicrosoftDocs), 'base' (master), 'head repository' (v/docs-help-pr), and 'compare' (pr-processing). A green checkmark indicates 'Able to merge'. Below this, a text area contains the commit message: 'Updated the create a pull request article with pull request processing information (from major workflow article). Created includes for this article as well.' There are buttons for 'Write' and 'Preview', and a rich text editor toolbar. A note at the bottom says 'Attach files by dragging & dropping, selecting or pasting them.' At the bottom right are 'Create pull request' and 'Allow edits and access to secrets by maintainers' checkboxes.

6. The commit message that you entered earlier is the default PR title. Edit the title to make it more appropriate for a PR, for example: *fix typo in <article name>*.
7. Clear the description box.
8. Select **Create pull request**. The new PR is linked to your working branch in your fork. Until the PR is merged, any new commits you push to the same working branch in your fork are automatically included in the PR.

## Pull request processing

The previous section walked you through the process of submitting your changes, through a new pull request. A pull request asks that your changes in the working branch be pulled and merged into another branch. In most cases, that other branch is the default/master branch in the main repository.

Each pull request undergoes a set of validation and auto-labeling to help repo admins. The pull request validation process is described below.

### Validation

Before your pull request can be merged, it will pass through one or more PR validation processes. Validation processes vary depending on the scope of the changes and the rules of the destination repository. After your pull request is submitted, you can expect one or more of the following to happen:

- **Mergeability:** A baseline GitHub mergeability test occurs first, to verify whether the proposed changes in your branch are in conflict with the destination branch. If the pull request indicates that this test failed, you must reconcile the content that is causing the merge conflict before processing can continue.
- **Contributor License Agreement:** If you're contributing to a public repository and aren't a Microsoft employee, depending on the amount of change, you might be asked to complete a short Contribution License Agreement (CLA) the first time you submit a pull request to that repository. After the CLA step is cleared, your pull request is processed.
- **Labeling:** Labels are automatically applied to your pull request, to indicate the state of your pull request as it passes through the validation workflow. For instance, new pull requests might automatically receive the "do-not-merge" label, indicating that the pull request hasn't yet completed the validation, review, and sign-off steps.
- **Validation and build:** Automated checks verify whether your changes pass validation tests. The validation

tests might yield warnings or errors, requiring you to make changes to one or more files in your pull request before it can be merged. The validation results are added as a comment in your pull request for your review, and they might be sent to you in e-mail.

- **Staging:** The article pages that are impacted by your changes are deployed to a staging environment for review upon successful validation and build. Preview URLs appear in a PR comment.
- **Automerger:** The pull request might be automatically merged, if it passes validation testing and certain criteria. In this case, you don't need to take any further action.

## View Acrolinx report

If the repository you're working in is set up for automated Acrolinx runs, within minutes you'll get an **Acrolinx Scorecards** email. For the article you changed, you'll see a numeric **Score**. If this score is below 80, you may not be able to merge the PR, depending on how Acrolinx rules are set up for your repository. Select the link in the **Scorecard** column to look at the error messages. The Acrolinx scorecards can be accessed from with the Pull request validation message.

## Review changes on the staging site

The next email notification you get is the publishing system reporting a successful publish. Select the link in the **Preview URL** column. The staging link is in the review.docs.microsoft.com domain and has a query string that identifies the Git branch. For a PR, the branch name is typically *pr-en-us-0000* (ends with PR number). Here's an example staging link: <https://review.docs.microsoft.com/en-us/help/contribute/contribute-create-a-pull-request?branch=pr-en-us-3790>.

Verify that content formatting has no problems. If there are formatting issues, and you don't know how to fix them, get help from someone familiar with docs Markdown.

The Docs Build validation message with the Preview URL can be accessed from with the Pull request validation message.

### TIP

You don't have to rely on email notifications; the same information is available in the PR page in GitHub:

- If your repo runs Acrolinx automatically, the PR page will have sections titled **Acrolinx Scorecards**, with links to the Acrolinx reports.
- The publishing system adds comment boxes with titles like **Validation status: passed**. Under the title is a list of changed files with **Preview URL** links.

## Update the pull request

After all PR processing is completed, you should review the results (PR comments, preview URLs, and validation) to determine if more changes are required before you sign-off for merging. If a PR reviewer has reviewed your pull request, they can also provide feedback through comments if there are outstanding issues/questions to be resolved before merge.

### To update the PR

When you're making changes, you almost always have to add commits to the PR:

- To address Acrolinx issues.
- To address rendering issues that appear on the staging site.
- To address feedback from peer reviewers.
- To address feedback from PR reviewers (certain repositories).

1. Switch back to Visual Studio Code. Ensure you are in the working branch from where you made the initial commit. See the bottom left of the VSCode window.



2. Open the file in VS Code to edit and make your changes.
3. Select **View** and choose **Terminal**.
4. For a list of all the files changed, run the following command:

```
git status
```

5. To stage the file with updates

```
git add <filepath> or git add .
```

6. To ensure all the right changes are stages

```
git status
```

7. Commit the changes to your pull request. Remember to include a message capturing the changes briefly.

```
git commit -m "commit message"
```

8. Push the changes

```
git push origin working branch name
```

Each time you add a commit to the same working branch in your GitHub fork, the commit is automatically added to the PR. And the publishing system automatically reruns Acrolinx and restages the affected article(s).

## Sign-off and merge pull request

Remember, your pull request has to be merged by a PR reviewer before the changes can be included in the next scheduled publishing run. Pull requests are normally reviewed/merged in the order of submission. If your pull request requires merging for a specific publishing run, you will need to work with your PR reviewer ahead of time to ensure that merging happens prior to publishing.

After your contributions are approved and merged, the docs.microsoft.com publishing process picks them up. Depending on the team that manages the repository you are contributing to, publishing times can vary. Articles published under the following paths are normally deployed at approximately 10:30 AM and 3:30 PM Pacific Time, Monday-Friday:

- <https://docs.microsoft.com/azure/>
- <https://docs.microsoft.com/aspnet/>
- <https://docs.microsoft.com/dotnet/>
- <https://docs.microsoft.com/enterprise-mobility-security>

It can take up to 45 minutes for articles to appear online after publishing. After your article is published, you can verify your changes at the appropriate URL:

```
http://docs.microsoft.com/<path-to-your-article-without-the-md-extension> .
```

## Automated labels

By assigning the appropriate label to a pull request, comment automation lets read-level users (users who don't have write permissions in a repo) do write-level actions. If you're working in a repo that has comment automation turned on, you can use hashtag comments to assign labels, change labels, or close a pull request. Comment automation will also notify Microsoft employees by email for review and sign-off of public repository PRs, whenever someone proposes changes to articles where you're the author.

HASHTAG COMMENT	WHAT IT DOES	REPO AVAILABILITY
#sign-off	When the author of an article types <b>#sign-off</b> in the comment stream, the <b>ready-to-merge</b> label is assigned. This label lets the reviewers in the repo know when a pull request is ready for review/merge.	Public and private
#sign-off	If a contributor who <i>isn't</i> the listed author tries to sign off on a pull request in a public repo, comment automation writes a message to the pull request indicating that only the author can assign the label.	Public
#hold-off	If authors change their mind or make a mistake, they can type <b>#hold-off</b> in a PR comment to remove the <b>ready-to-merge</b> label. In the private repo, <b>#hold-off</b> assigns the <b>do-not-merge</b> label.	Public and private
#please-close	Authors can type <b>#please-close</b> in the comment stream to close the pull request if they decide not to have the changes merged.	Public
#label:"custom label text"	Authors can add a custom label up to 200 characters (shorter recommended).	Public and private

When the pull request is free of errors and signed off, your changes are merged back into the parent branch and the pull request is closed.

## Merge the PR

Typically, when your PR is ready to go, you get it reviewed before it's merged. Ask for a review from a team member who is familiar with the docs editing process for your repository. Continue to merge the PR when the reviewer approves the PR.

1. Use the repository-specific method for getting a PR merged. For example:

- In the Azure repository, add a PR comment **#sign-off**.
- In the .NET and ASP.NET repositories, select the **Squash and merge** button.

After the PR is merged, your changes will be included in the next publish.

2. After the next publish, go to the production URL for the article you changed, and verify that your changes are live.

## Merge conflicts

While you're working on an article, someone else might make a change to the same article. If their change gets

merged into upstream main/default before yours, there is potential for a merge conflict that will prevent your PR from being merged. In that situation, one way to resolve the conflict is to [rebase](#) your working branch on the current upstream main/default branch.

You can rebase at any time—you don't have to wait for a merge conflict to happen. Whenever you suspect someone else might have made conflicting changes, rebase your working branch.

## Next steps

Now that you know how to create pull requests, learn how to get them reviewed:

[GitHub for content reviews](#)

# Best practices for contributing to docs.microsoft.com

5/24/2021 • 6 minutes to read

When authoring docs in external repositories (for example, `azure-docs-pr`), familiarize yourself with these best practices.

## Working with GitHub

### Make changes on the private repo

If you are making changes on GitHub's web-based Markdown editor, make sure that you have navigated to the private repository (for example, <https://github.com/MicrosoftDocs/azure-docs-pr>) and not the public repository (<https://github.com/MicrosoftDocs/azure-docs>). Note the ending of `-pr` in the correct repo.

Add the [SpineEdit extension](#) to Chrome and Chromium Edge and your changes will be made **automatically in the correct location**.

#### TIP

Why is the private repo so important?

- The build process in the private repo validates links, TOC, formats, and more.
- The public repo goes through external teams before it is triaged to the appropriate content team.
- It's faster to publish on the private repo.
- The private repo allows you to preview your changes in the docs system.

### Don't make changes on your master branch

Don't make changes on your master branch. Unlike the internal repo, your new branch is located under your personal fork and can therefore be named according to any convention you like.

```
git checkout -b <new branch name>
```

### Keep your master branch up to date

Pulling upstream syncs your local default with the docs default branch, which is the absolute truth. If you don't sync often enough, you risk getting merge conflicts with recently updated files. Pull upstream in either of the following cases:

- At the beginning of your work session
- Every time before creating a new branch

```
git checkout master  
git pull upstream master
```

## Authoring tips and Markdown

### Markdown shortcuts

Use Alt+M to open the Markdown wizard in Visual Studio Code.

### Metadata

All external articles contain a header with metadata. Make sure your metadata is correct and update if necessary.

- ms.date: MM/DD/YYYY
  - Update the date when you make significant content changes.
- author: author GitHub alias
- ms.author: author MS alias
- msreviewer: article reviewer (MS alias) - could be content developer or SME

#### NOTE

- The listed author receives all alerts on doc changes and GitHub issues, SEO fixes, as well as communication with PR reviewers and editors.
- The person responsible for maintaining the article should be listed as the author and ms.author.
- If you have authored an article, you should add yourself as author and ms.author or msreviewer so that you:
  - get credit
  - are consulted about future changes

### Raise your Acrolinx score above 80

Acrolinx is a plug-in that assists you to edit your texts according to Microsoft standard. It automates feedback on brand, terminology, grammar, spelling, punctuation, style, and voice. Most repositories require a score of 80 to publish. Preferably, aim for a score of 90 or more, so that it will be more difficult for future edits to bring down the Acrolinx score to below the passing grade of 80.

- GitHub: You can use Acrolinx in GitHub. Acrolinx results are reported in the Conversation tab within a box titled Acrolinx Scorecards in your PR. Acrolinx scores each article with every new commit pushed to the same PR.
- Visual Studio Code: You can use Acrolinx locally in Visual Studio Code. Many contributors prefer this option since they get feedback before submitting their PRs, and the experience is more interactive. Set up Acrolinx in Visual Studio Code with [these instructions](#).

#### NOTE

To enable Acrolinx in Visual Studio Code you must open the entire data-explorer folder in your workspace, and not a single file.

### Preview your article

There are two ways to preview the formatting of your article.

- In the toolbar of Visual Studio Code, click on the preview button or the side-by-side preview button:



- In the build report of your PR. The build report is found at this web address: <https://github.com/MicrosoftDocs/dataexplorer-docs-pr/pull/XXX> where XXX is your PR number. Under the Preview URL column in the validation status window, select View

## Validation status: passed

File	Status	Preview URL	Details
<a href="#">data-explorer/kusto/query/topnestedoperator.md</a>	 Succeeded		

## Use correct text formatting

Use [this guide](#) to know when to use bold, italics, and code styles.

## Use correct image naming and formatting

- [How to create a screenshot](#)
- [File naming tips for images](#)

## Link responsibly

- Use internal links to other docs within the same repo. An internal link should look like this:

```
[link text](markdownfile.md)
```

- Links to articles in other Docs repos or docsets are known as site-relative links. For example, to link to the article at <https://docs.microsoft.com/windows/uwp/get-started/>, the link syntax is /windows/uwp/get-started/ as shown here:

See [Get started with Windows 10 apps](#)

Within our own repo, a site-relative link looks like this:

See [Azure Resource Manager template](#)

- Use a complete URL to link to an external site, including non-docs.microsoft.com Microsoft sites. All links must be secure (https instead of http) whenever the target supports it.
- Don't include locale codes such as en-us in your links to docs.microsoft.com, MSDN, TechNet, and Azure.com articles.
- Don't use *click here* as your link text. It's bad for SEO and doesn't adequately describe the target.
- Only links within the same repo are verified by the build process. Check your other links.

### NOTE

If you're using Visual Studio Code, you can use the markdown wizard to enter relative and external links by typing Alt+M

## Choose everyday words

See [this list](#) on the contributor's guide about replacing formal words with everyday words

## Change the TOC when you add or delete files

The TOC file is typically located in the same folder as your articles (for example, azure-docs-pr\articles\backup\toc.yml).

- If you delete a file entirely, either remove its link from the TOC or replace it with a more updated file.
- Deleted files must have a redirect entry. Any deleted file needs a redirect to some location, even if its specific instance is removed from the TOC.
- If you add a new file, add its entry to the correct location in the TOC in the repository.

## Add a redirect

Deleted files must have a redirect entry in the redirect file (for example `.openpublishing.redirection.json`).

Redirect entries include the following information:

- `source_path`: The source path of the deleted file. This is a markdown file name.
- `redirect_url`: A site-relative link to the new location.
- `redirect_document_id`: `false`

### Redirect entry example

```
{  
  "source_path": "articles/iot-accelerators/iot-accelerators-device-simulation-create-custom-device.md",  
  "redirect_url": "/previous-versions/azure/iot-accelerators/iot-accelerators-device-simulation-create-  
custom-device",  
  "redirect_document_id": false  
},
```

## PR and review process

### Understand the review process

Your documents will go through several review steps, each of which improves the quality and consistency of the docs set. In general, the review steps are:

1. Review by the appropriate content team member - the author of the article is automatically listed as a reviewer. In order to speed up this process, you can send the reviewer a message via email or Teams.
2. `#sign-off` - you must manually enter this exact phrase in the comments section.
3. Review by PR team (see [PR team schedule](#))
4. Possible request for more changes - you will be notified via GitHub email notification if relevant (unless you have disabled notifications in GitHub settings).
5. Merged into master - done manually by the PR reviewer team.
6. Publish to live - automatic.

### Tag your reviewer and send them a message

Once you have committed your changes and opened a PR on GitHub, if you are working with a content team, let them know. Most of the GitHub notification emails they get look the same. Do the following steps to best communicate what you've done:

- Mention the content team reviewer by typing `@GitHubUsername`. If you don't know their GitHubUsername, see GitHub aliases.
- Send them a message on teams and make sure this PR is on their radar.

### #Sign-off

When you or the content team has determined that the content is ready to merge, it's time to sign off.

1. Navigate to the PR website (for example, <https://github.com/MicrosoftDocs/azure-docs-pr/pull/XXX> where XXX is the number of your PR).
2. **Make sure that the build process has finished. You can't sign off on an in-process build.**
3. At the bottom of the **conversation** tab, add a comment containing the phrase `#Sign-off`. You can also tag other contributors or add general comments at this time.
4. Select **Comment** and not **Close and comment**

**NOTE**

At this point, your work is complete.

# Helpful resources for contributors to docs.microsoft.com (DMC)

3/5/2021 • 2 minutes to read

Listed below are the resources available to contributors to docs.microsoft.com for support and guidance on using our platform. If you're unsure about which resource to use, please see the [Quick Decision Chart](#) at the bottom of this page to help you decide where to go for help.

## SiteHelp

<https://sitehelp.microsoft.com>

Sitehelp is the one-stop shop for getting immediate response for your p0 or p1 issue. If some part of our infrastructure is broken, if you see a bug, or you have a rendering problem, use <https://SiteHelp>. Sitehelp is backed by an [SLA](#) for production level incidents and for pre-production level bugs.

## Docs Support

[Docs Support](#) is a channel in Microsoft Teams that the docs.microsoft.com engineering team hosts. Coverage for this channel rotates daily, with both an engineer and a PM will be covering responses.

Use this channel if you can't find help in the Contributor guide, or get stuck, or if you're having a specific issue with your repository or branch that you need help with.

## Git or GitHub help

Use the [Docs Support > General channel](#) in Teams. Add the tag `@Git` along with details of your issue (platform, repo, PR #, etc.). @mentioning `@Git` messages a group of git help volunteers to assist in answering your question.

## Publish-on-docs

Use <https://aka.ms/publish-on-docs> to request operational support from our vendor teams that can help you get your repo configured right.

## Contributor Guide

Review the [Docs Contributor Guide](#) to find task-based information and guidelines for all Microsoft contributors to docs.microsoft.com.

## Contributor Training

Learn to contribute anywhere on docs.microsoft.com. To schedule a training session, fill out form at [Content Contributor Central](#).

## Quick Decision Chart

Use this chart to help figure out which resources to turn to when you need help when contributing to docs!

MY SITUATION	THE FIRST PLACE TO TURN
Something is supposed to work, but is broken, and it's <b>visible to customers</b>	<a href="#">Sitehelp</a>
An internal tool isn't working correctly	<a href="#">Sitehelp</a>
specific problem with your branch or repo	<a href="#">Contributor Guide</a>
specific problem with your branch & you <b>already</b> searched the contributors guide	<a href="#">Docs Support</a>
Builds are very slow	<a href="#">Docs Support</a>
Request a docs feature	<a href="#">Docs Feature</a>
Need a new repo	<a href="#">Publish-on-docs</a>
Need to set a repo live	<a href="#">Publish-on-docs</a>
Need to learn how to publish on docs	<a href="#">Docs Training</a>
GitHub isn't working correctly	<a href="#">Docs Support</a>
Need to set up public contributions	<a href="#">Publish-on-docs</a>
Need to change base URL or priority	<a href="#">Publish-on-docs</a>
Want help with improving content performance	<a href="#">Docs Training</a>
Need a release branch	<a href="#">Publish-on-docs</a>
Need to set a repo live the first time	<a href="#">Publish-on-docs</a>
Need help with generating redirections	<a href="#">Publish-on-docs</a>
Need help with setting up conceptual versioning	<a href="#">Publish-on-docs</a>
Need to make a config change	<a href="#">Publish-on-docs</a>

# Quick reference for contributors to docs.microsoft.com (DMC)

5/4/2021 • 5 minutes to read

This quick reference is a filtered view of the [Contributor's Guide](#) for Advocates, Program Managers, or anyone else who is not a full-time Content Developer.

IF YOU WANT TO...	THEN...
Contribute for the first time	<a href="#">Configure your GitHub account</a>
Make a small change (for example, fix a typo)	Click the <b>Edit</b> button on a published article; change the <i>repo name</i> by editing the URL. <i>For example</i> , from <a href="#">docs-help</a> to <a href="#">docs-help-pr</a> or from <a href="#">azure-docs</a> to <a href="#">azure-docs-pr</a> and submit fixes. The <a href="#">SpineEdit browser extension</a> can make this process even easier.
Locate an author of a specific article	Click the <b>Edit</b> button at the top of a published article, the article will open within the GitHub user interface; click on the <b>Raw</b> button; find the GitHub username of the author from the <code>author</code> value or the domain alias from the <code>ms.author</code> field.
Notify an author of a problem	Email the article author.
Submit changes to an article	Submit a pull request in the private repository from your fork. <i>For example</i> , <a href="#">docs-help-pr</a> or <a href="#">azure-docs-pr</a> from <a href="#">your fork*</a> ; be sure to @ mention the doc author in the PR
Create a new branch	While on the main/default branch such as <a href="#">docs-help-pr</a> or <a href="#">azure-docs-pr</a> , run: <code>git checkout -B &lt;BRANCH_NAME&gt;</code> . This will create a local branch on your local fork of the remote repository. Note that remote branches originating from <a href="#">azure-docs-pr</a> are release branches, and are <a href="#">managed</a> .
Sync your branch with latest changes	While on your branch, run: <code>git pull upstream master</code> . For this command to work you need to have a remote named <i>upstream</i> that points to the private repo such as, <a href="#">docs-help-pr</a> or <a href="#">azure-docs-pr</a> .
Copy branch to GitHub for a PR	Run: <code>git push origin &lt;BRANCH_NAME&gt;</code> . Once the branch is copied to GitHub, you can navigate to the web app to create a PR.
Request to publish a PR	Add <code>#sign-off</code> to a comment in your PR
Find out when my merged update will be live	Check the <a href="#">PR review and publishing schedule</a> for your repo. You can also read about <a href="#">publishing from a public repo</a> .
Submit a PR to the public repository	Please don't. Use the private repository to get the full benefit of build-time validation of your content. <i>For example</i> , <a href="#">docs-help-pr</a> or <a href="#">azure-docs-pr</a>

IF YOU WANT TO...	THEN...
Create a GitHub issue	Please don't create public or private issues in GitHub to report problems with content. For the time being, email the author with your feedback.

\* This action is against the private repositories such as [docs-help-pr](#) or [azure-docs-pr](#).

## FAQ

### Why use the private repo?

Pull requests against the private repo benefit from a series of build tasks that ensure content quality. Feedback from these tasks helps you write better content, plus you get a preview link to verify changes before you publish. Using the private repo allows you to have conversations in a private without accidentally sharing information not appropriate for public consumption.

Refer to [Should I work in a private or a public repository?](#) for more detail.

### When should I submit changes to the public repo?

Avoid making PRs and creating issues against the public *docs-help* or *azure-docs* repositories. There are two reasons for this restriction:

1. Issues and PRs opened against the public repo are subject to an SLA from our CXP team.
2. The build system for the private repositories includes a number of validation steps to ensure content quality.

### How do I add a new article to the docs?

There are two facets involved with writing docs: [environment setup](#) and content planning.

When it comes to planning, it's important that you meet with a Content Developer to make sure your ideas fit with the current content model and plans. Contact the Content Developer responsible for the area you are interested in to craft a plan.

#### TIP

For best results meet with a Content Developer before you begin writing your content.

### How do I find the Content Developer responsible for the content I want to write?

There are a few ways you can approach finding the right Content Developer:

- Find an article that is close to what you want to write and email the author. This person will either help you directly or route you to the right person.
- Ask the Engineering team who the Content contacts are
- Advocates can access the [AirTable database](#) for a list of Content contacts

### What should I expect to discuss with a Content Developer?

As you collaborate with a Content Developer, look to discuss topics like:

- How does this idea fit into the existing content plan?
- Who is responsible for content and technical reviews for your content?
- Who is going to maintain the content long-term?

### Do I really have to fork this 10 GB+ repository for a small change?

No. For small changes, or changes isolated to just one Markdown file, you can use the GitHub online editor to submit a PR against [docs-help-pr](#) or [azure-docs-pr](#).

The [SpineEdit browser extension](#) can make this process easy.

If you opt to use the GitHub online editor manually, make sure to change the *repo name* in the URL from *docs-help* or *azure-docs* to [docs-help-pr](#) or [azure-docs-pr](#) before making changes.

Refer to [Edit an article on Docs](#) for more detail.

## What ways can I collaborate with Content Developers?

Opportunities to collaborate are far and wide:

- Want some subject-matter-expert advice on a conference talk?
- Perhaps you have a series of videos that you'd like to see surface in the docs?
- Maybe you'd like to help provide feedback "from the trenches" on an Azure service?
- Think you can help take an existing set of docs to the next level?

All these opportunities (and more) are perfect places to create connection points with the Content team.

## What happens when I submit a PR?

Once you create a PR, the build process begins and:

- Your content is validated against writing standards
- Preview versions of changed files are generated

Once the PR is error-free and you want to publish your changes, add a comment that includes `#sign-off` in order to notify a PR reviewer of your request.

The reviewer is responsible to help keep content consistent with publishing standards. Reviewers may ask you to make some changes or just merge your branch. After all issues are resolved and your branch is merged into master, your changes become public the next time content is pushed live.

## Resources

We offer the [Visual Studio Code Docs Authoring Pack](#) as a collection of extensions that helps support writing standards, gives you access to templates, allows you to preview content. Before you publish content use the [feedback from Acrolinx](#) to help improve the quality of your content.

The following links take you directly to areas of the [Contributor's Guide](#) that will help you write your articles:

### Writing

The format is a bit different for each type of content.

- [Overview](#)
- [Quickstart](#)
- [Tutorial](#)
- [Sample](#)

### Images/Screenshots

There are a few guidelines to follow regarding screenshots and images.

- [Screenshots: How to create, format, and embed in documentation](#)
- [Create an expandable screenshot \(lightbox\)](#)
- [Create conceptual art](#)

### Video

Videos for the docs need to account for localization and closed captioning.

- [Plan a video](#)

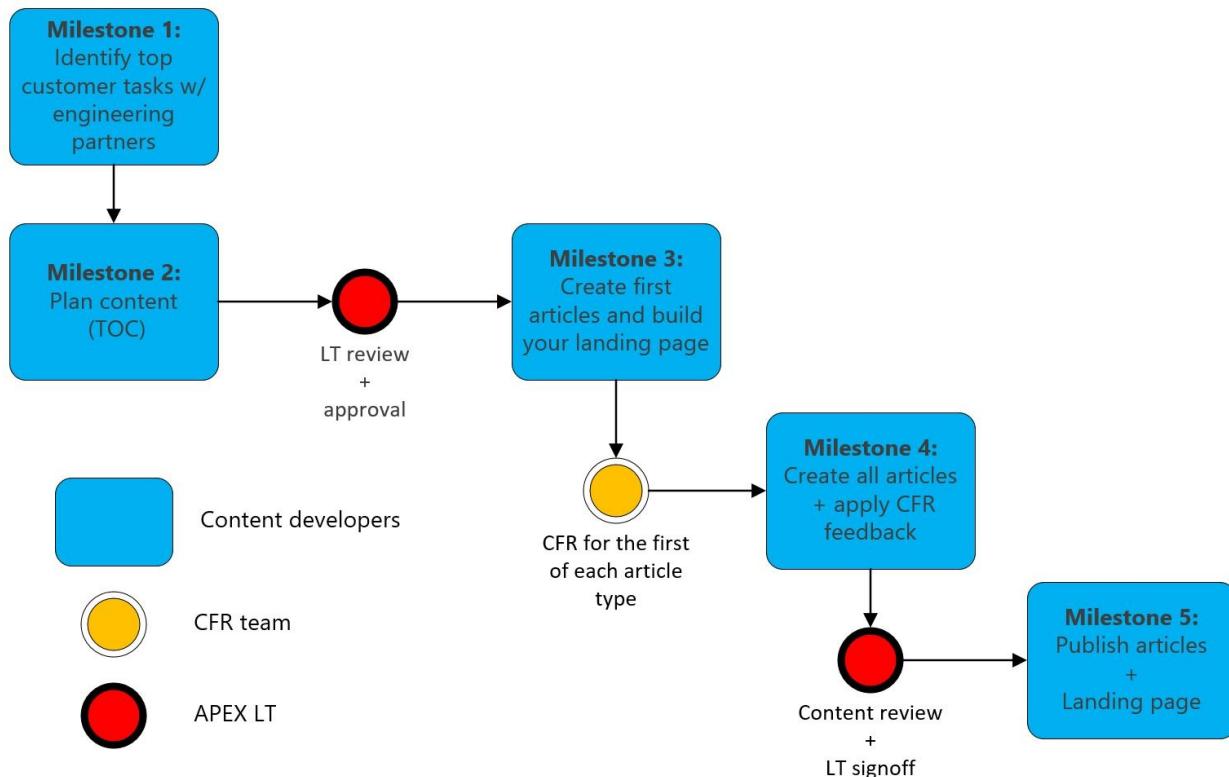
- [Create a video](#)
- [Publish a video](#)

# Writing content for C+AI

5/10/2021 • 3 minutes to read

The process for publishing content on docs.microsoft.com was created to support content quality and consistency. It starts with direct collaboration with your engineering partners to define what customers want to do with your product and continues through five milestones.

The existing PR approval and publishing processes still apply following milestone 5.



TASK	DELIVERABLE	REVIEWERS	APPROVERS
<b>Milestone 1: Top tasks</b>  Work with your engineering partners to identify the top 10-15 tasks and use cases customers want your service or technology for. These are user tasks, not feature descriptions. They state, in customer language, what customers want to accomplish with the service.	Approved list of use cases and common tasks	Writing team manager, writers	Content & Learning LT

Task	Deliverable	Reviewers	Approvers
<p><b>Milestone 2: Plan your content and your TOC</b></p> <p>Build your content plan using the <a href="#">TOC checklist spreadsheet</a>. Include the tasks and a customer intent statement for each article.</p> <p>Identify your plan for emphasizing the CLI in procedure articles (quickstarts, tutorials, how-to guides). If you can't provide procedures using the CLI, document why not in the TOC checklist.</p> <p>Review the existing content to see what you can revise to support your top tasks. Identify areas where you will need new content, as well as articles to delete. A <b>definition for what type of content belongs in each section of the TOC is in the <a href="#">Guidance for TOC structure</a> article.</b></p> <p>Work with your writing team M2 manager for content plan/TOC approval.</p>	TOC checklist	Writing team M1 manager	Writing team M2 manager

Task	Deliverable	Reviewers	Approvers
<p><b>Milestone 3: Create your first articles</b></p> <p>Set up a release branch and create a single overview, quickstart, tutorial, and sample for one platform or one language. The goal is to get the quality right before moving on to the rest of the articles. Use the correct templates when they exist, and apply Content &amp; Learning writing guidelines.</p> <p>Build your <a href="#">landing page</a></p> <p>When you start writing your articles and landing page, you may identify the need for an exception to the existing content criteria. If that happens, exceptions need approval by Martin Ekuan. Identify, and get approval for, your exceptions early. Any exceptions that are not part of the review and approval after milestone 4 will block or slow publishing.</p> <p>When CLI and PowerShell scripts can be written to automate common tasks, write them and store them in the repo.</p>	Content review	Writing peers	Content & Learning LT for exceptions
<p><b>Milestone 4: Create all articles for a release</b></p> <p>Create the rest of the articles in your content plan that apply to a planned release, including the overview article, a comparison article (if needed), quickstarts and tutorials, and CLI/PS scripts or code samples.</p> <p>Only write articles where the platform story is complete. Apply feedback from Milestone 3 review as appropriate to these articles.</p> <p>Update the TOC and landing page.</p>	Release branch corresponds to approved content plan	Writing team M1 Manager	Writing team M1 Manager

Task	Deliverable	Reviewers	Approvers
<b>Milestone 5: Publish</b> Publish all planned content. Keep the comprehensive content experience in mind. Think about customers as they move through your articles.	Final approval prior to publish	Writing team, Engineering partners	Content & Learning LT prior to publish

## Update content

Because the content model continues to evolve, you should check for changes implemented since you last worked on an article and make them the next time you update that content.

### **Adding new overview, quickstart, or sample articles**

These require review/approval by your manager.

### **Adding new tutorials**

The articles in the Tutorials node of a TOC must align with top customer tasks identified in milestone 1 (and approved via the TOC checklist review in milestone 2)

After the original content refactoring effort is complete, new content in the Tutorials node could be either:

- New content for a top customer task already approved: If a new tutorial is for a top customer task that was already approved during review of the TOC milestone checklist, no additional review is required.
- New content for a new task: If a new tutorial is for a top customer task that has not yet been approved, the updated customer task list must be reviewed and approved by the M2.

Considerations for the review:

- Is the new task really a top customer task?
- Does it replace something already on the list (triggering the work to move the demoted task to the How-to node of the TOC)?

## Content goals

- Articles that offer a clear, best path to success for customers.
- A process that creates less content, but keeps us focused on the 20% of content that addresses 80% of what customers want to use the product for.
- A portfolio we can easily maintain for relevancy and technical accuracy.
- A consistent customer experience across docs.microsoft.com

## How do I get started?

- [Write an overview](#)
- [Write quickstarts](#)
- [Write tutorials](#)
- [Create a landing page](#)

# Contributing content to docs.microsoft.com

4/15/2021 • 2 minutes to read

There are five main phases of work that go into the development of effective content. These steps apply to all content on docs.microsoft.com. This article presents an overview of the content creation process and a brief description of each phase.

Contributors drive the process for their content, but some tasks are done in partnership with others.

## Planning

The planning phase helps a contribution effort start effectively by properly identifying the customer and clearly understanding the scope of content needed to complete the work. Typically the planning phase develops a clear understanding of the customer's needs, which results in a clear content plan. This plan can be reviewed and approved by all stakeholders. You also identify critical milestones and deadlines during planning. Doing this work first sharpens the focus on what content needs to be designed so wasted effort is minimized.

## Designing

Once your customer needs and content expectations are established in the planning phase, contributors should consider the preliminary design of the content.

When designing content, consider how the customer might find the content (SEO). Consider also what the structure of the content will look like, and similarities and dependencies with existing content.

## Developing

Writing is the main phase of the content development process. Although it can take some time to write content, proper planning and design can accelerate this phase. The writing phase produces a draft. The phase also completes with relevant media files and folders, error-checked style. Stage changes in a release branch in preparation for a pull request.

## Reviewing and publishing

Once the content is drafted, you open a pull request (PR). You add reviewers to the pull request so they can provide feedback. The PR also triggers build validation, which checks for technical errors such as broken links. Some content also receives feedback with stylistic suggestions from Acrolinx.

Once feedback and validation issues are addressed, the PR is signed off and merged with the main branch.

## Maintaining

Maintenance is the final phase of the content development process. It is also an independent and ongoing phase. The content you create should be monitored indirectly, through metrics or customer feedback, as well as directly, by content review for up-to-date references and relevance.

## Next steps

[Plan your content](#)

# Plan your content

4/15/2021 • 4 minutes to read

This article describes the tasks of the planning phase of content contribution. It's a part of a series of articles that help you develop and maintain quality content.

Planning is the first phase of the content development process. It begins when you identify a need for content. This need can be based on upcoming releases or current gaps in content.

You can contribute content effectively with good planning and these basic ideas:

- Clearly understand content scope and accurately identify the customer.
- Empathize with stakeholder perspectives early on. Alignment with stakeholders ensures that content appropriately meets reviewer expectations and reduces iterations in development.

## Content goals

- Collaboration on Docs content is more appealing if it's easier.
- An article should offer the clear, best path to success for its customers.
- Our process should create less, focusing on the 20 percent of content that deals with 80 percent of what customers want.
- A portfolio succeeds when we can easily maintain its relevancy and technical accuracy.
- The customer experience should remain consistent across docs.microsoft.com.

## Identify and confirm stakeholders

You might already know your stakeholders from previous work. Make sure that you've determined your best points of contact and that they have confirmed support. Usual stakeholders include the lead program manager, the executive sponsor, and subject matter experts. Work with them to set publication timelines and a content plan.

When you define goals, set regular meetings (for example, weekly or biweekly) with the product owner. Product owners can tell you what features are coming down the line. Some features might have particular user-facing implications. Think about examples, corner cases, and gotchas. Product owners can also tell you which developers worked on a product. You can communicate with them by email or chat for any questions.

## Identify customers

Understand customers' values and challenges. Consider intersections with other initiatives, such as marketing. The better you can describe your content, the more prepared (or qualified) you'll be to draft it.

Beyond product features, prepare a list of the use cases or top tasks of your customers. How customers use the product might differ from what you're most familiar with or responsible for. "Complete and correct" documentation might not be complete or correct if the ultimate audience isn't the intended one. Documenting these gaps sometimes take precedence over documenting features, based on severity.

Think about when a customer might search for your content. What do they need to get from it? Write from there.

[Generate a customer intent statement](#) that stakeholders and content developers agree on. Try to express your

point in the customers' language. For more information, see [Writing principles job aid](#).

## Scope and coordinate content

Documentation development can occur in tandem with, or linked to, other project work streams such as testing or training. But documentation development style can vary.

*Planning-as-development* is a live collaboration development model that some prefer. Project managers know what's coming down the line. With this foresight, and the experience of working through the execution, it makes sense for PMs to generate first drafts. These drafts are used in coordination with developers to self-validate and clarify steps in development.

Some light efforts might not require explicit planning, if the documentation process, product, and teams are in place. Be wary of unplanned investigating homework that's sometimes involved with accurate documentation!

In some cases, documentation might come late in the process, in a "release readiness" sense. This communication and material are similar to, and line up with, product promotion and support. If the release is a public preview, review how it will affect your [table of contents](#) and landing page.

Whether you document first, last, or as you go, ensure that the expected timeline is well communicated and understood. Determine key elements to emphasize and what forms of documentation are most pertinent (for example, quickstarts, tutorials, or how-to guides).

## Review existing content

You can do a few things to make sure your content doesn't duplicate content that's already published:

- Examine relevant libraries.
- Search the content repository (repo).
- Search the platform or organization.
- Search for key terms.

Review the existing content to see what you can revise to support your top tasks. Identify areas where you'll need new content, and identify articles to delete. Content development plans can be tracked in Azure DevOps, Excel, GitHub, Teams, Planner, or Outlook (for example).

## Planning tasks

The following table summarizes tasks that you should perform in the planning phase.

TIP	
You can do many of these tasks at the same time or in no particular order, which can simplify or speed up the planning phase. Links in the task column go to detailed task documentation for that entry.	
	TASKS
<input checked="" type="checkbox"/>	Identify the lead PM, sponsor, and subject matter experts for the new content.
<input type="checkbox"/>	Identify the expected timeline for the release.
<input type="checkbox"/>	Identify if a repository for your need exists. If not, <a href="#">request the creation of a content repo</a> .

	TASKS
<input checked="" type="checkbox"/>	Work with your product team to understand customer challenges or user scenarios to help determine the content that you want to create. Review the <a href="#">writing principles</a> and the <a href="#">acceptable content for Microsoft Docs</a> .
<input type="checkbox"/>	Consider whether you'll need <a href="#">reference content</a> for any APIs, libraries, packages, or CLIs. Onboarding times can be long, so take that into consideration.
<input type="checkbox"/>	Determine if the release is public preview or general availability. For a preview, see <a href="#">Identify preview content</a> , and review how it will affect your table of contents and landing page.

## Next steps

[Design your content](#)

# Design your content

4/15/2021 • 6 minutes to read

Now that you've finished [planning](#) your content rollout, it's time to design the content. Design is the second phase of the content development process. Content design helps break down various tasks into executable chunks.

In the content design phase, you'll identify resources and services that you need to help communicate and deliver the information. You'll also align all the tasks in a flow that simplifies the next phase: content writing.

This article identifies and defines the tasks in designing content. It's one of a set of articles that help you develop and maintain high-quality content.

When you're designing content, consider how the customer might find the content (search engine optimization, or SEO). Think about the structure of the content, along with the similarities or dependencies with other content.

## Align your scenario to customer needs

Writing fewer articles that focus on the real issues that customers face creates a better experience than writing a lot of content that might, or might not, be useful. Focus on the story you want to tell the customer. What content will land that message effectively?

Use the [writing principles and voice](#) to draft a customer intent statement that addresses customer needs for your documentation. This approach will help to communicate and organize the content efficiently.

## Determine the content structure

After you identify the most important tasks, you'll need to define the structure of the content. This structure includes a list of articles, how the content will flow, and what the access points are for the content. Defining what you want your customers to do with your product or service will help better position your content.

Based on your content needs, list the articles that you want to create. Choose the [article type](#) that will be the correct format for the information in each article. Be sure to follow the [acceptable content](#) guidance.

### Table of contents

A [table of contents](#) (TOC) is a visual map of the content structure. It provides an overview of the covered topics and allows for easy navigation to the content of choice.

Define the TOC structure based on the list of articles that you've identified. TOCs are created in the [.yml file format](#).

### Breadcrumbs

While a TOC provides the primary navigation, a breadcrumb serves as a secondary navigation. A breadcrumb helps identify the location, path, or hierarchy of content.

If you're adding or updating articles within an existing content set, breadcrumbs have been established already. To create breadcrumbs for a new content set, review the [breadcrumb guidelines](#) and [instructions for building a breadcrumb file](#) based on your article list and TOC structure.

### Landing pages and hub pages

Start thinking about a [landing page](#) and [hub page](#) for your content.

A landing page is an entry point for your customers into your content set. A hub page provides a big-picture view of how your product or service is aligned with other products and services. A hub page encompasses a broader set of products and is one level above the landing page.

[Submit a request](#) to add the new content on your area hub page (if it applies) and include the new icon to use.

## Review cross-service content

It's common for a content set to have touch points with other services, content collections, or learning paths. While you're designing content, find any dependencies across services or products to better align a contribution with the existing body of content. It's key to manage both the people and content portions during such dependencies.

From a project management perspective, you need alignment with all stakeholders to keep the project in the priority list. Adhering to timelines also becomes essential to keep the project moving forward.

In a scenario of cross-service dependencies, use links to reference existing content. For example, use [includes](#).

## Review terms and SEO

Customers will discover your content if you emphasize the language that they use when they search for answers to their questions. This language often includes the issues they're facing and not product names.

This is where your earlier work on identifying the real customer needs will help your writing. Research and identify commonly used [terms](#) across products and services. They bring consistency to the user experience. Ensure that your usage doesn't mean the opposite for another service.

You'll need to establish [metadata values](#) (if they don't exist already), which become a part of your service slug. The service slug is the part of the docs.microsoft.com URL that identifies your service. The slug will be a part of the ms.service attribute in articles.

Considerations for terminology in your content design include:

- Will this content be [localized](#)?
- Will terms used in the content translate well to other languages?

### Search engine optimization

[Optimizing content](#) for search engines requires understanding how customers find your content. Discoverability of content requires careful curation of the terms and [keywords](#) to be used in the content.

SEO doesn't end with some terms sprinkled all over the content. It's a habit that every content contributor develops. SEO is applied in [titles](#), [metadata](#), and [H1s](#).

### Metadata

Metadata describes the various facets of a piece of content in consumable bits of data. Metadata, when set up properly, makes content easy to find, use, and manage. Metadata values drive all our content reports, discoverability metrics, and content consumption metrics.

Before you [request new metadata values](#), review the [required metadata](#) values that you can incorporate in your content.

## Design checklist

The following table puts the preceding tasks in a checklist:

<input checked="" type="checkbox"/>	TASKS
<input type="checkbox"/>	<p><i>If a repo doesn't exist:</i> Work with your product team and marketing to determine the service slug value. The service slug is the part of the docs.microsoft.com URL that identifies your service. The service slug is used for the repo folder or subfolder name. It's the value of the ms.service metadata attribute in articles.</p> <p>For example, <i>virtual-machines</i> is the service slug for Azure Virtual Machines and is consistently used between the Docs site (<a href="https://docs.microsoft.com/azure/virtual-machines">https://docs.microsoft.com/azure/virtual-machines</a>) and the ACOM site (<a href="https://azure.microsoft.com/services/virtual-machines/">https://azure.microsoft.com/services/virtual-machines/</a>).</p>
<input type="checkbox"/>	<p><i>If a repo already exists:</i> Work with the sponsor to coordinate your new content.</p>
<input type="checkbox"/>	Create a <a href="#">new release branch</a> for your new content.
<input type="checkbox"/>	<p>By using the new service slug value, create a corresponding folder to house your new content. Create the folder locally, and add your first files. The folders are "created" in GitHub when you add the first file and push it to the online repo.</p> <p>For example, this folder matching the service slug is used for the Azure Virtual Machines articles:  <a href="https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines">https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines</a>.</p>
<input type="checkbox"/>	<p>Create the <a href="#">new metadata value</a>. Use the <a href="#">new ms.service value request form</a> to add the new metadata to the list. Be sure to select <b>Yes</b> to include your new Docs metrics in the <a href="#">content engagement reports</a>. The ms.service metadata property typically uses the same value as the service slug.</p>
<input type="checkbox"/>	Get the new .svg icon from the marketing representative for your release or product.
<input type="checkbox"/>	Based on your content needs, list the articles that you need to create. Review the recommended <a href="#">article types</a> for Docs.
<input type="checkbox"/>	Based on your article list, <a href="#">define the TOC structure</a> to develop the toc.yml file. If a related toc.yml file is available, copy it into your new content folder and change it based on your article list. Use the <a href="#">TOC checklist spreadsheet</a> to draft your TOC.
<input type="checkbox"/>	Start thinking about <a href="#">landing pages</a> and <a href="#">hub pages</a> for your content. If a similar index.yml (landing page) file is available, copy it into your content folder and change it to suit your needs.
<input type="checkbox"/>	Add your content set to the main breadcrumb file: \bread\toc.yml. Review the <a href="#">breadcrumb guidelines</a> .

	TASKS
<input checked="" type="checkbox"/>	If you have a product forum URL, add it to the docset's docfx.json file: MicrosoftDocs/docs-help-pr/docfx.json. Under <code>feedback_product_url</code> , add your folder path and feedback URL. (See other line items as examples.) Review the article <a href="#">How to enable the documentation feedback control</a> .
<input type="checkbox"/>	<a href="#">Submit a request</a> to add the new content on your area hub page (if it applies), and include the new icon to use.

At the end of the design phase, you'll be clear on how to address customer problems along with product or service features. You'll have an idea of what the content will cover, how it will be structured, and how it will be discovered. Having these items configured will make the next step, [content writing](#), focused and efficient.

## Next steps

[Write your content](#)

# Writing your content

4/15/2021 • 2 minutes to read

This article identifies and defines the tasks in content writing. It is one of a set of articles that are designed to help you develop and maintain quality content.

The purpose of this phase is to compose the main body of your contribution in preparation for review.

Writing is the main phase of the content development process. Although it can take some time to write content, proper planning and design can accelerate this phase. The writing phase completes when a draft, including relevant media files, is written and checked for style errors. Stage the working content draft in a release branch for a pull request.

## Content writing tasks

### TIP

Links in the task column below go to detailed task documentation for that entry.

~	TASK	DESCRIPTION
□	<a href="#">Create a release branch</a>	Set-up a GitHub account and create a release branch in the upstream private repo. Share its name with all authors collaborating on content release. The typical naming convention is <i>release-</i> prefix followed by feature or content name.
□	<a href="#">Write code</a>	Think about how to lead a user through creating a proof of concept and developing a piece of code. Ensure that you test the code and it performs as expected.
□	<a href="#">Write first draft of content</a>	Configure your authoring environment, set your remote repositories, and create a local branch. Then, create your Markdown files for your content.
□	<a href="#">Create rich media</a>	Create images and media effectively and accessibly. Save your media in a <code>media</code> folder, and reference it with correct Markdown syntax.
□	<a href="#">Address style errors</a>	Run an Acrolinx check within VSCode and ensure the Acrolinx score for each article is at least 85.

## Next steps

[Review and publish your content](#)

# Publish your content

4/29/2021 • 3 minutes to read

This article describes the tasks of content publishing. It's one of a set of articles that help you develop and maintain high-quality content.

## Manage your pull request

After you've created your content, primarily .md ([Markdown](#)) files, you create a pull request (PR) in [GitHub](#).

Your PR automatically starts build validation checks. Whatever checks run will depend on your repo configuration. The Docs repo has validation for [style](#) and [compatibility](#). This validation checks grammar or broken links, for example.

Address any validation warnings or build errors in your PR. For a validation warning, the validation comment includes details to guide you on what to fix. For build errors, you can get more information by selecting the link to the build report.

### Content review

Before customers see your content, it needs an adequate review and approval by subject matter experts, content owners, or other stakeholders. It's helpful if your PR has no build errors or warnings before you invite collaborators.

If you have write permissions for the repo, you can add reviewers through the GitHub UI. If appropriate, you can notify reviewers in a comment by @-mentioning their GitHub alias. In your comment, be clear about your expectations for the review. Depending on the people that you include, you might get an editorial pass on the content or its structure, or you might get technical feedback.

Some reviewers might not be available to fully review a submission. As long as you tag relevant parties, everyone has a chance to chime in if they see any glaring issues. If you've [planned properly](#), reviewers won't be surprised with a notification for review.

### Final review and merge

After your PR passes validation and you address any feedback, you sign off on the PR by using a [#sign-off](#) comment. It then goes through a [final review with a vendor team](#).

When the reviewers at this stage are satisfied that your content meets established criteria, they merge the PR. Your content merges with the main branch of your team's repo and is ready for your customers to use.

#### NOTE

If you're using a [release branch](#) for your content, that release branch should merge with the primary branch of the appropriate Docs repo. The timing should be as planned for your project. It should align or be in sync with [Go Live](#) or other milestones for your content set. Your team might have made these timing decisions during the planning or design phase for your product, service, or feature.

## Check your content after publication

Between the various tools in the publication pipeline, it's possible to introduce a broken link, misformatted text, or simple typos. After your content goes live, read it once more to ensure that it's clean and complete.

Be mindful that debugging or troubleshooting any problems, such as why a redirect isn't working or why a change isn't appearing in the samples browser, can take time.

If you're updating a landing or hub page, consider the possibility of bumping out stale content, rather than adding a significant body of content to what might already be a busy page.

## Summary of content publishing tasks

The following table summarizes tasks that you might expect in the publishing phase. Links go to detailed documentation for that entry.

<input checked="" type="checkbox"/>	TASKS
<input type="checkbox"/>	When your content is ready, understand the <a href="#">pull-request (PR) submission recommendations</a> and create a PR to receive feedback from SMEs.
<input type="checkbox"/>	Check the build validation, and address any issues.
<input type="checkbox"/>	Check the Acrolinx results, if available. See <a href="#">Acrolinx coverage</a> .
<input type="checkbox"/>	<a href="#">Invite reviewers</a> to provide feedback on the new content.
<input type="checkbox"/>	After you incorporate the reviews and feedback, sign off ( <a href="#">#sign-off</a> ) on your PR to merge it. If you're using a release branch, you might create several PRs to bring in content from all contributors. Review each PR before merging with the release branch. Signing off is a requirement for Microsoft Docs. Talk to your repo admin for your process.
<input type="checkbox"/>	Schedule your <a href="#">release branch merge</a> with the main branch.
<input type="checkbox"/>	<a href="#">Request Go Live</a> .
<input type="checkbox"/>	Generate a <a href="#">forward link (FWLink)</a> to your landing page so your product team can link to the content from their product's user interface.
<input type="checkbox"/>	Check if your content requires a Product Launch Readiness (PLR) sign-off. PLR is a process where the product and the articles require a sign-off for the public preview or GA stage. For Azure, remember to request a <a href="#">PLR sign-off</a> for articles.
<input type="checkbox"/>	After the content is published, do a visual pass of all the content, including any hub and landing pages.

## Next steps

After you publish your content, be sure to maintain it well.

[Maintain your content](#)

# Maintain your content

3/11/2021 • 3 minutes to read

The purpose of the maintenance phase is to ensure that published content continues to meet customers' needs. Customers expect our documentation to be clear, fresh, and technically correct. You need to continually review and update documentation to keep up with product changes, address customer feedback, and ensure positive customer engagement.

The best way to reduce future maintenance is to invest the time up front to plan, design, and write high-quality documentation. Do your best the first time around: have a clear customer intent, use the right topic type and structure, and write technically accurate content.

Maintenance is an active phase with the same tactical scope as any other phase in the [content development process](#). Underperforming content can damage the customer experience and the Docs brand.

## General maintenance

*General maintenance* refers to the straightforward tasks for keeping content fresh. You should do it as much as is reasonably possible. At a minimum, maintain articles that you yourself created. (Remember to retire or combine content to control volume.)

General maintenance tasks include:

- General freshness review:
  - Adding content to address new features
  - Adding or replacing screenshots for new UI
- Adding or updating links for new URLs or resources
- Updating names and terminology
- Addressing customer feedback
- Making basic edits (for example, fixing grammar, formatting, layout, and metadata)

### TIP

The `date` metadata attribute is a general measure of article freshness.

## Qualitative maintenance

*Qualitative maintenance* is updating content based on a quality measure like customer feedback. We need to act on feedback whenever possible. If we don't act on it, customers will stop giving it.

### GitHub issues

If a customer wants to suggest an improvement to content, they might create a [GitHub issue](#). An improvement might be fully formed, such as a user providing an updated link or screenshot. The issue might also be a suggestion: the customer might not know the answer, but the current documentation is inaccurate or lacking. For more information, see [Triaging new GitHub issues](#).

### Public pull requests

Sometimes, internal content teams must work with external contributors to determine how to resolve content

suggestions. For best-practice guidance on handling community contributions, see [How to work with public pull requests](#).

## Verbatims

Customers might submit feedback through a [verbatim](#). Feedback and comments can be directly actionable for content owners, localization teams, and site engineering teams.

## Quantitative maintenance

*Quantitative maintenance* is related to customer-engagement issues that we find in ways other than direct customer input.

Whether content receives feedback doesn't tell us whether the content is effective. Content might not receive feedback or views for various reasons.

Your content might have other issues. For example, it might not be targeted or detailed enough, or maybe users can't find it in the first place. To understand how customers engage with your documentation, see [Data, BI, and reports overview](#).

For more information about quantitative maintenance, see:

- [GitHub issue and pull request report](#)
- [Documentation and reference report](#)

## Continuous improvement

The decisions that you make while [planning](#) or [designing](#) one article can influence the planning or design of another article, or several in a set about a common product or service. Propagating such design changes--for example, across all articles for a consistent look or feel--might not add enough value to prioritize the effort over other team initiatives.

The task of applying Microsoft voice principles, standardizing, or otherwise improving articles is a continuous effort, so it's called *continuous improvement*. These improvements can include:

- [Retiring](#), [renaming](#), or [archiving](#) old content. Outdated content can make a customer spend time looking for answers in the wrong places. If content is for a product or feature that's no longer supported, or it's no longer valid and won't be fixed soon, it might be best to archive or to retire the content.
- [Optimizing](#) content. You can optimize existing content by identifying ways to increase its value to the customer. For example, is there a clear customer intent that's reflected in the title? Is the article discoverable through search? Can you make the article more concise? Is it scannable? Should you add art? Are links clear and do they go to the right place?
- [Validating](#) [freshness](#). Make sure that technical content is up to date. Content that changes often includes product names, UI, scripts or code snippets, samples, and region availability.

## Checklist of maintenance tasks

The following table summarizes tasks that you should perform in the maintenance phase.

	TASKS
<input checked="" type="checkbox"/>	Validate accuracy. The longer content has been published on the live site, the more likely it needs a review for accuracy. Dependencies, brand names, and metadata can change over time.

	TASKS
<input checked="" type="checkbox"/>	Review content performance.
<input type="checkbox"/>	Address any <a href="#">GitHub issues</a> .
<input type="checkbox"/>	<a href="#">Retire old content</a> .

## Next steps

[Contribute to Docs](#)

# Choose the correct topic type for your article

3/10/2021 • 2 minutes to read

You can use this list of standard topic types to help identify the correct format for the information you want to provide. Some of these topic types have [ms.topic attribute values](#) associated with them. When you write a new article make sure you're using the correct metadata.

TOPIC TYPE	WHAT IS IT	WHEN TO USE IT
Overview	<p><b>Required:</b> One article explaining the service from a technical point of view.</p> <p><b>Optional:</b> A second article comparing features across related services/projects.</p>	For new customers. To explain the service, technology, or functionality from a technical point of view. It isn't intended to define benefits or value prop; that would be in marketing content.
Quickstarts	<p><b>Recommended:</b> Fundamental day-1 instructions for new customers to quickly try out a specific product/service.</p> <p>The focus is on getting the product/service into the hands of customers so they can use it, not on documenting the procedure for setting it up.</p>	When you can get the service, technology, or functionality into the hands of new customers in less than 10 minutes. Quickstarts are preferred, but not required if no meaningful functionality can be introduced to new users in under 10 minutes.
Tutorials	<p><b>Required:</b> Scenario-based procedures for the top customer tasks identified in milestone one.</p> <p>The focus is on showing the customer how to do the task, not on helping them set up their own environment.</p>	To show the single best procedure for completing a top 10 customer task.
Samples	<p><b>Recommended:</b> Example scripts that align to the most common tasks. The scripts should be short, simple, and quick to complete.</p>	To provide the most common scriptable tasks using PowerShell, CLI, Azure Resource Manager templates, or code samples.
Concepts	<p><b>Optional:</b> In-depth explanation of functionality related to a service(s) that are fundamental to understanding and use.</p>	When you want to: <ul style="list-style-type: none"><li>- explain what something is and define its critical functions.</li><li>- describe how something works or what happens when something is done.</li><li>- provide guidelines for completing a job task in varied situations.</li></ul>

Topic Type	What Is It	When To Use It
How-to guides	<b>Optional:</b> Procedural articles that show the customer how to complete a task.	To provide the steps for completing a task.  To help customers complete tasks in their own environment. How-to guides differ from tutorials in that they can include optional information, explanations, and information to help inform decisions.
Troubleshooting articles	<b>Optional:</b> Articles that help users solve a common issue.	To help resolve a specific error or problem that customers commonly run into and would search on.
Reference	<b>Optional:</b> Documentation for APIs, PowerShell cmdlets, CLI commands, or other types of language-based content.	To provide descriptions of things like settings, values, keywords, and configurations. These topics are where readers will go to look up a specific value.
Resources	<b>Optional:</b> Non-technical content customers may find useful as they use the product/service.	To provide links to supporting information like pricing, Stack Overflow, blogs, regional availability, Azure Video Library.

# Write an overview

5/10/2021 • 3 minutes to read

The overview article talks about the product/service from a technical point of view. It's not intended to define the benefits or value prop; that just duplicates marketing.

The three questions an overview article needs to answer are:

- What is it?
- Why should I care?
- How can I get started?

Use [this template](#) when writing a new overview article.

You may also create a second article that compares features across related services. For example, see [Comparing hosting options](#).

## Overviews in the Table of Contents

The Overview section of the TOC for your product or service is limited to two articles; a required overview, and an optional comparison.

TOCs must be consistent across products/services. Don't come up with new TOC nodes on your own. Unless it's unavoidable, make sure your TOC labels never wrap to a second line.

- **Overview** - Use the TOC entry **Overview** as a heading that expands. This entry shouldn't link to any article itself. This section should not be expanded automatically (expanded: false).
  - **What is X?** - To link to the main overview article, use a TOC entry in the form of a question, such as "What is X?" where the service name is X.
  - **Compare** - The optional comparison doc should follow as the second TOC entry.

Filter by title

Service Bus Messaging Documentation

▼ Overview

**What is Service Bus Messaging?**

        Compare messaging services

    > Quickstarts

    > Tutorials

    > Samples

    > Concepts

    > How-to guides

    > Reference

    > Resources

Example YML code for these TOC nodes:

```
- name: Overview
  items:
    - name: What is Service Bus Messaging?
      href: overview.md
    - name: Compare messaging services
      href: comparison.md
```

## Metadata

Make sure the following metadata is included in your overview article.

ATTRIBUTE	VALUE
ms.topic	overview

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The comment format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This comment provides a record of the intent of this overview for future contributors.

## Templates

Use this [standard overview template](#) unless there's a more product/service/technology appropriate template in the repository-specific section of the guide.

## H1 (Headline)

For the H1 - that's the primary heading at the top of the article - use the format **What is <service>?** You can also use this format in the TOC if your service name doesn't cause the phrase to wrap.

The H1 has the second greatest impact on search rank and relevance. The H1 isn't the same as the page title in search (found in the metadata section).

## Introduction

The introductory paragraph helps customers quickly determine whether an article is relevant, and can be reused as the meta description.

Describe in customer-friendly terms what the service is and does, and why the customer should care. Keep it short for the intro.

You can go into more detail later in the article. Many services add artwork or videos below the introduction.

See how this article introduces the service:

### [App insights](#)

## Body

After the intro, you can develop your overview by discussing the features that answer the "Why should I care"

question with a bit more depth.

Be sure to call out any basic requirements and dependencies, as well as limitations or overhead.

Don't catalog every feature, and some may only need to be mentioned as available, without any discussion.

## Screenshots and videos

Screenshots and videos can add another way to show and tell the overview story. But don't overdo them. Make sure that they offer value for the overview. The engineering team may have a good video introducing the service. If it's short, clear, and engaging, you can embed it.

Your first screenshot should always include the **full browser window in an Edge window**.

With all art and videos, make sure that as the service is updated, the art and video remain pertinent. **If they fall out of sync, remove or replace them.**

For more information, see [Screenshots: How to create, format, and embed in documentation](#).

## Next steps

In Overview articles, provide at least one next step and no more than three. Use regular links; don't use a blue box link. What you link to will depend on what is really a next step for the customer, such as:

- Do they need to set up or register their environment before they can use the service?
- Do you have a quickstart or tutorial that they dive into?
- Is there a developer reference they can browse?

# Write quickstarts

5/28/2021 • 11 minutes to read

Quickstarts are fundamental day one instructions that help new customers quickly use a product or service. The emphasis is on getting the functionality into the customer's hands, so they can experience it.

- The entire activity is a short set of steps that provide an initial experience.
- The audience is customers who are new to the product, service, technology, or scenario.
- The experience should ideally take no longer than 10 minutes for the customer to complete. Wait time for things like provisioning doesn't count against the 10-minute limit.

Quickstarts are preferred, but not required if no meaningful functionality can be introduced to new users in under 10 minutes. If you have no quickstarts, leave that node out of the TOC.

Use [this template](#) when writing a new quickstart article.

If your quickstart involves an Azure SDK client library, see the more specific guidance in [Quickstarts for Azure client libraries](#).

This guidance will help you make sure your quickstart meets the latest publishing criteria. For an understanding of the overall content development process, see [Writing content for Content & Learning](#).

## Quickstart checklist

Here's a quick checklist to make sure your quickstart meets the key formatting guidelines.

# Quickstarts

<Service name>
Documentation
> Overview
✓ Quickstarts
① <customer task> - Portal
<customer task> - PowerShell
<customer task> - Azure CLI
> Tutorials
> Samples
> Concepts
> How-to guides
> Reference
> Resources

The screenshot shows a quickstart article structure. It includes:

- Step 2: An H1 section.
- Step 3: A "Prerequisites" section with a bullet point about an Azure account and a link to create one.
- Step 4: A screenshot of a Microsoft Azure browser window showing the "Sign up for Azure Policy" page.
- Step 5: A "Clean up resources" section.
- Step 6: A "Next steps" section with a blue button containing a link.

NUMBER	REQUIREMENT
1	Titles in the TOC must not wrap to a second line.
2	Start your H1 with "Quickstart:" and try to keep the H1 under two lines on a 1920x1080 screen.
3	Make the first H2 "Prerequisites" and include a link to create a free Azure account if applicable. See the template for the standard text.
4	If users access your product/service via a web browser, include the browser frame in the first screenshot in a quickstart. (Don't use a screenshot in the introduction, however; save it for later.)
5	Clean up resources (H2) should come just before Next steps (H2)
6	Next steps (H2) heading is required. A single link in the blue box format should direct the customer to the next quickstart or first tutorial.

The checklist above covers key guidelines. See the [standard quickstart template](#) for complete details. The

following sections also provide additional details for specific areas.

## TOCs

If you have a quickstart article, locate it in a Quickstart node in the TOC. More information about TOC requirements is available [here](#).

Avoid word wrapping quickstart titles in the TOC.

The Quickstarts section of the TOC is expanded by default. If there's no Quickstarts section, the Tutorials section should be expanded by default.

Create (and include in the TOC) a quickstart for each of the common methods (for example, portal, command-line language) or for the frameworks/languages supported by the service. For example:

COMMON METHODS	OR	FRAMEWORKS/LANGUAGES SUPPORTED BY THE SERVICE
Quickstarts - Create VM - Portal - Create VM - CLI - Create VM - PowerShell		Quickstarts - Create app - .NET - Create app - Java - Create app - JavaScript - Create app - Python

### NOTE

We're in the process of creating guidelines for use of zone pivots and conceptual tabs to combine related variants into a single article. If you're interested in that process, contact Erik Hopf and Kraig Brockschmidt.

## Templates

Use the [standard quickstart template](#) unless there is a more product/service/technology appropriate template in the repository-specific section of the guide. For example, quickstarts for SDK client libraries should use the [Quickstarts for Azure client libraries](#).

## File name

If you are repurposing an existing article, consider keeping the existing file name unless it's misleading. That way you avoid redirects and keep the data about the article.

## Metadata

Make sure the following metadata is included in your quickstart.

ATTRIBUTE	VALUE
ms.topic	quickstart

### Title attribute

Include the word "Quickstart" in the title attribute.

### Description attribute

Include the word "quickstart" in the Description attribute. "In this quickstart, you..."

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
# Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

Documenting this statement provides a record of the intent of this quickstart for future contributors.

## H1 (Headline)

The primary heading of the article should be in the form, "Quickstart: <do something with X>", that is, make the heading a statement of action using a particular service or technology along with a specific framework or language. For additional details, see the [standard quickstart template](#).

Don't add "Quickstart:" to the H1 of any article not in the Quickstarts node of the TOC.

## Introduction

Begin a quickstart with the following text: "Get started with [service/product] by using [specific tools or activities] to [achieve a goal or job to be done]. [Summary of steps]. [Brief statement of cost incurred.]".

This pattern focuses on the job or task that the reader completes in the context of the service or product and applicable tools. The goal is to specifically sell the *article* rather than sell the service, although the description of the job or goal can include a little of the service's capabilities. The focus is on what the reader learns and tries to answer the question, "What is this article about?"

For example: "Get started with Azure Functions by using command-line tools to create a function that responds to HTTP requests. After testing the code locally, you deploy it to the serverless environment of Azure Functions. Completing this quickstart incurs a small cost of a few USD cents or less in your Azure account."

When describing tools and services, use general industry terms (such as "serverless," which are better for SEO) rather than Microsoft-branded terms or acronyms (such as "Azure Functions" or "ACR").

### TIP

If you'd like some help crafting the introduction, contact Kraig Brockschmidt ([kraigb](#)).

In the introduction to a Quickstart, avoid the following items:

- Green checkmark lists, checklists of subsections, or lists of learning goals. Those structures are only for tutorials.
- Lists of service capabilities.
- Callouts: if there are useful notes for the reader, include them later in the article.
- Screenshots or diagrams, because they typically push the prerequisites out of view. You can use these later in the article in an appropriate place.
- Links, except for links to alternate versions of the same content. For example, if an article is focused on a Visual Studio Code experience it's appropriate to link to a CLI experience because it helps readers get to the right content. Otherwise links distract the reader from completing the quickstart.

## Prerequisites

The first H2 of the quickstart should be "Prerequisites", even if there aren't any prerequisites. Having the heading in all cases creates consistency between Quickstarts.

The section indicates what a customer needs to address before they start:

- If there aren't any prerequisites, just place "None" in the section.
- Otherwise, list prerequisites as a bulleted list of *items*, not instructions. Items typically should link to the appropriate installers or secondary instructions. If there isn't a suitable link target and the requirement has more than only simple instructions, consider making the prerequisite a procedural step.
- See the [next section](#) about standard text for the first item if it's an Azure account.

If you feel like your quickstart has too many prerequisites, the quickstart may be the wrong content type: a tutorial or how-to guide may be the better option.

If you feel like a requirement needs instructions, make a step in the quickstart for that procedure.

See the [standard quickstart template](#) for additional details including specifying items, linking, and the preferred ordering of items.

### Create an Azure account

Because quickstarts are intended to help new customers quickly use a product or service, the first requirement is an Azure account. Use the following text:

- **Azure**

- An Azure account with an active subscription. [Create an account for free]([https://azure.microsoft.com/free/?WT.mc\\_id=A261C142F](https://azure.microsoft.com/free/?WT.mc_id=A261C142F)).

For Azure, don't provide instructions or links to the "try it now" experiences. The "try it now" experiences are for marketing content; documentation should link to the free trials that can be easily converted to paid accounts.

Note that the text above uses "account" deliberately because research has shown that "account" is better understood and approachable than "subscription." It helps readers understand that a subscription is associated with an account. Furthermore, the link on [azure.microsoft.com](https://azure.microsoft.com) uses "account."

- **PowerApps**

- A PowerApps account. [Create an account for free](<https://web.powerapps.com/signup>).

- **Power BI**

- A Power BI account. [Create an account for free](<https://app.powerbi.com/signupredirect>).

### Prerequisite check (optional)

#### Preview

User research has shown that not having an appropriate environment is a common reason why people fail to complete a quickstart.

For this reason, consider an H3 under Prerequisites named `### Prerequisite check` to validate the environment. In that section, use a simple bullet list with the necessary validation commands or instructions. For example:

- In a terminal or command window, run `func --version` to check that the Azure Functions Core Tools are version 2.7.1846 or later.
- Run `az --version` (Azure CLI) or `Get-Module -ListAvailable AzureRM` (Azure PowerShell) to check that the Azure CLI version is 2.0.76 or later.
- Run `az login` (Azure CLI) or `Login=AzureRmAccount` (Azure PowerShell) to sign in to Azure and verify an

active subscription. \*\* Run `py --version` (Windows) or `python --version` (MacOS/Linux) to check your Python version reports 3.7.x.

## Cloud Shell include

For quickstarts that use the Azure CLI and PowerShell, test the procedure using Azure Cloud Shell. If you can run all the CLI or PowerShell steps in Cloud Shell, then you can consider using the following /include file as a step after the prerequisites.

```
[!INCLUDE [cloud-shell-try-it.md](../../../../includes/cloud-shell-try-it.md)]
```

If you can't run all commands in the Cloud Shell, don't use the /include file. Instead, specify Azure CLI or Azure PowerShell in the Prerequisites (specifying the appropriate version number)

- For the Azure CLI

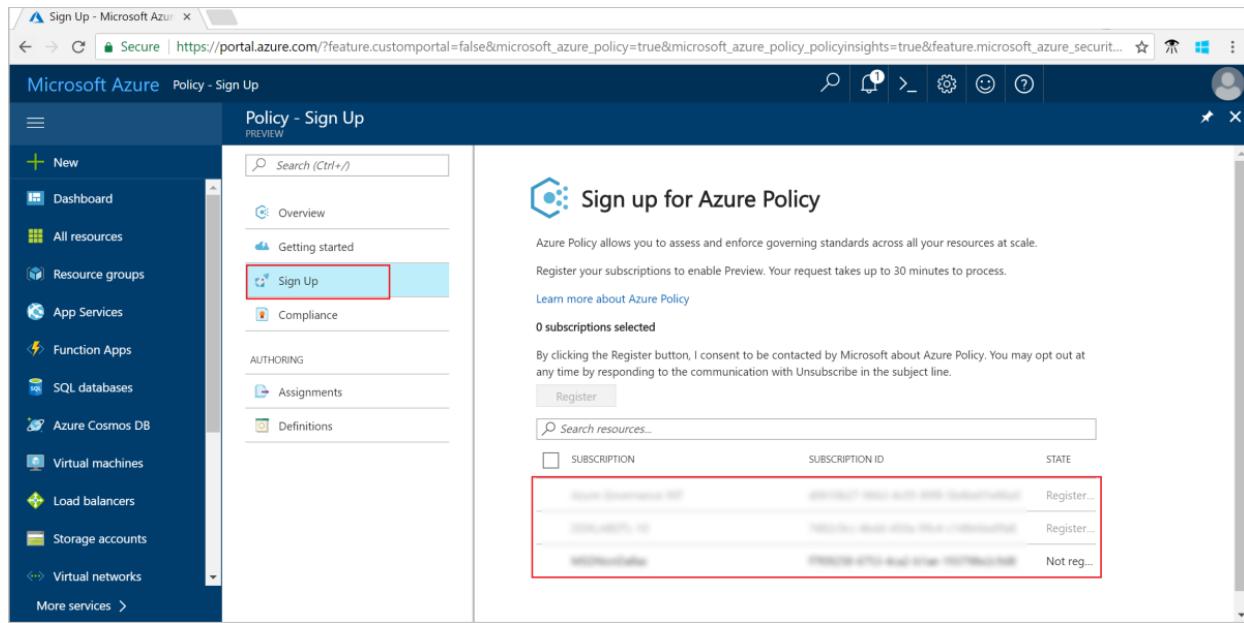
```
- [Azure CLI 2.0.76+]( /cli/azure/install-azure-cli).
```

- For Azure PowerShell

```
- [Azure PowerShell 5.1.1+](https://docs.microsoft.com/powershell/azure/install-az-ps)
```

## Screenshots

If users access your product/service via a web browser, the first screenshot should always include the **full browser window in Chrome or Safari**. This perspective is to show users that the portal is browser-based and OS agnostic.



Make sure that additional screenshots offer value for the customer (explaining complex UI, providing context, validating they've taken the correct steps or are in the right place, etc.). Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article.

For more information, see [Screenshots: How to create, format, and embed in documentation](#).

## H2s and H3s

Don't number H2 and H3 headings. Quickstarts are procedure-based and customers understand that the

sections follow in sequence.

## Linking from the procedure

Quickstarts are intended to provide the customer with a single prescriptive path to success. Don't link off to other content in the procedural part of the quickstart, so the customer doesn't have to leave the article.

An exception is where you know from user research that readers typically want more information in a particular place, especially a reference link. Links to non-reference content (for example, conceptual articles) are seldom useful in such contexts.

## Alerts

Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.

If you need to use alerts, limit them to one or two per article. Multiple notes should never be next to each other in an article.

## Code blocks

Code requires specific formatting. Here are a few useful examples of commonly used code blocks. Make sure to use the interactive functionality where possible.

Here is an example of a code block for Java:

```
cluster = Cluster.build(new File("src/remote.yaml")).create();
...
client = cluster.connect();
```

or a code block for Azure CLI:

```
az vm create --resource-group QuickstartCreateVM-rg --name myVM --image win2016datacenter --admin-username
azureuser --admin-password myPassword12
```

or an interactive code block for Azure PowerShell:

```
New-AzureRmContainerGroup -ResourceGroupName QuickstartCreateVM-rg -Name mycontainer -Image
mcr.microsoft.com/windows/servercore/iis:nanoserver -OsType Windows -IpAddressType Public
```

As with the Cloud Shell, use the -interactive code fence variants only if **all** commands can be run in the interactive shell. Otherwise, the shell won't have all the state from previous commands and will likely fail in a confusing manner. Put simply: if you can't use -interactive for all commands, don't use -interactive for any commands.

### Use a specific resource group name

When instructing the reader to create a resource group in Azure, suggest a name that's specific to the quickstart, such as "QuickstartCreateVM-rg" or "AzureFunctionsQuickstart-rg", and be sure to include the conventional "-rg" suffix (a best practice). If the resource group might be used in follow-on articles, use a name that makes sense in those articles as well.

Using a specific name keeps the resources created in this quickstart separate from those created when following unrelated quickstarts, especially those in other services. The specific name makes it easy for the reader to clean up the resources from this specific article.

## IMPORTANT

This guidance overrides previous guidance that suggests using "MyResourceGroup" everywhere. Using a generic name can cause unexpected collisions with existing resources, which pose obstacles to successfully completing a quickstart.

## Clean up resources

If the customer creates any Azure resources in the article, include an H2 entitled **Clean up resources** (before **Next steps**) that tells the customer how to delete those resources and avoid incurring any ongoing costs. You can invite customers to leave those resources in place, especially if those resources are needed for the next step article.

Ideally, all resources were created with a specific resource group name, in which case you can instruct the reader to delete that group on the portal, or to use `az group delete --name <name>` from the Azure CLI if the CLI has been used elsewhere in the article.

## Next steps

Finish the Quickstart with a **Next steps** H2 that points to the next logical article in a series using a blue button created with the following markdown:

```
> [&gt; [!div class="nextstepaction"]  
> [Azure Windows virtual machine tutorials](./tutorial-manage-vm.md)
```

Do not link to troubleshooting content or FAQs. Including this content implies the customer is going to run into known issues. Either fix these issues beforehand or create a different path to success for users that avoid these problems.

# Quickstarts for Azure client libraries

11/2/2020 • 8 minutes to read

A client library Quickstart helps a developer get up and running quickly *with the library*. Its primary goal is to help the developer install the library's package and perform several key coding tasks in their chosen development environment. A library Quickstart isn't intended to introduce a service and its features, though these concepts can be conveyed within the context of helping developers use the library to interact with the service. Think "code first."

If your Quickstart isn't specifically about helping a developer get started with an SDK client library, use the [standard Quickstart template](#) instead.

If the product team has created a [README.md](#) for the library, use it as a starting point for your Quickstart. This library Quickstart format is modeled after the Azure SDK README format, and much of the README's content can be reused here.

## IMPORTANT

The terms *SDK*, *client library*, and *package* have distinct meanings. See the [Terminology](#) section of the [Azure SDK Design Guidelines](#) for their appropriate use.

**Title:** The H1 of your Quickstart should be in the format:

```
# Quickstart: [Product Name] client library for [Language]
```

- Example: `# Quickstart: Azure Batch client library for Python`

**Introduction:** The introduction appears directly under the title (H1) of your Quickstart and serves to describe what the reader does in the article, rather than trying to describe the service.

- Sentence **one**: use the following structure, which is specifically crafted for SEO traction: "Get started with the [service/product] client library for [language] to [achieve specific goals]."
  - In the [achieve specific goals] phrase, describe what the quickstart demonstrates with the service. This phrase provides an opportunity to highlight capabilities of the service in the context of what the reader does in the article, rather than making separate marketing-type statements about the service.
  - Example: "Get started with the Computer Vision client library for Python to analyze a variety of visual features in an image and to recognize printed and handwritten text."
  - Contact Kraig Brockschmidt ([kraigb](#)) if you'd like help with crafting this opening sentence.
- Sentence **two**: use the following, which is also crafted for SEO traction: "Follow steps to install the package and try out example code for basic tasks."
- Sentence **three**: if applicable, include a brief statement about any costs one might incur by following the article. For example, "Completing this quickstart will incur on the order of USD \$2 in your Azure account".
- Lastly, include the following **single line of links** targeting the library's companion content at the bottom of the introduction; make adjustments as necessary, for example NuGet instead of PyPi:

[API reference documentation](#) | [Library source code](#) | [Package \(PyPi\)](#) | [Samples](#)

The markdown for this line is as follows:

[API reference documentation](<https://docs.microsoft.com/python/api/azure-cognitiveservices-vision-computervision/azure.cognitiveservices.vision.computervision>) | [Library source code](<https://github.com/Azure/azure-sdk-for-python/tree/master/sdk/cognitiveservices/azure-cognitiveservices-vision-computervision>) | [Package (PyPi)](<https://pypi.org/project/azure-cognitiveservices-vision-computervision/>) | [Samples](<https://azure.microsoft.com/resources/samples/?service=cognitive-services&term=vision&sort=0>)

In the introduction, avoid trying to sell the service; again, the introduction should sell the article itself.

Also avoid links (which may distract the reader away from the article) and avoid other conceptual content. You can place links and concepts later in the article in a context where it's likely (as shown from user studies, for example) that a reader might go off on their own search to answer a question (such as find the reference for a command or class).

If there is an important callout that applies to the article, you might include it after the line of links. Consider, though, whether the callout can be placed later in the article instead of the introduction.

## Prerequisites

Specify the requirements a developer must satisfy prior to beginning the Quickstart; for library quickstarts, installing the library is a step and not a requirement.

State the requirements as *items* rather than actions. Don't include instructional steps in this section; if instructions are needed, create a section for that procedure. Also include only those things the reader must have in place prior to proceeding with the quickstart. For example:

```
* An Azure account with an active subscription. [Create an account for free] (https://azure.microsoft.com/free/?WT.mc\_id=A261C142F).  
* [Python 3.6+](https://www.python.org/downloads/)
```

Which renders as follows:

- An Azure account with an active subscription. [Create an account for free](#).
- [Python 3.6+](#)

Do use the exact phrase above for Azure, which uses the familiar and well-understood term "account" and associates it with the less-familiar term, "subscription." We also use "account" because the link provided speaks of an "account."

When linking to installers or runtime versions, point to a specific downloads page rather than an informational page. The example above links to python.org/downloads rather than just python.org.

## Setting up

Walk the reader through preparing their environment for working with the client library. Include instructions for creating the Azure resources required to make calls to the service, obtaining credentials, and setting up their local development environment. For example, include **resource group** creation, **service account** (for example, Storage account, Batch account) creation, and any setup appropriate for their programming language (for example, a `dotnet new` command for .NET or `virtualenv` setup for Python).

- Include instructions for obtaining credentials and other requirements (like an endpoint URI) needed to instantiate an authenticated client object.
- Include Windows, MacOS, and Linux shell examples if you provide console-based instructions for creating resources, obtaining credentials, or setting environment variables. This typically means PowerShell and Command Prompt examples for Windows and Bash examples for MacOS and Linux.

- Note that PowerShell is the most popular shell among users of Azure documentation, followed by bash on MacOS. Try, then, to lead with Windows/PowerShell, followed by bash/MacOS. Linux comes last.
- Use Azure portal screenshots sparingly, and only if necessary. Screenshots take up valuable real estate and can cause a developer scanning your article to assume it's not a code-focused article, and bounce.

#### WARNING

Never instruct anyone to paste credentials directly into their code. Use environment variables or another authentication mechanism (such as Azure Key Vault) that keeps their credentials out of code.

## Install the package

Provide instruction for obtaining and installing the library's package. This section might include only a single line of code, like `pip install awesome-azure-library`, but should enable a developer to successfully install the package from NuGet, PyPi, npm, or Maven.

This section is **required** within the *Setting up* section, and must be named as specified. Try to push it as high up in the section as possible for better visibility. Simply seeing the package install command helps a developer know they're in the right place.

Again, use external links sparingly, and preferably not at all.

## Object model

Briefly introduce and describe the functionality of the library's **main classes**. Include links to their reference pages.

Explain the object hierarchy and how the classes work together to manipulate resources in the service.

## Code examples

Include code snippets and short descriptions for each task you listed in the introduction (the bulleted list). Briefly explain each operation, but include enough clarity to explain complex or otherwise tricky operations.

If available, present the same example snippets that the library's **README** includes. Such examples should be pulled from code within the library repository, and that code should be tested and verified as functional by continuous integration (CI).

Each example in the *Examples* section starts with an H3 that describes the example. At the top of this section, just under the *Examples* H2, add a bulleted list linking to each example H3. Each example should deep-link to the types and/or members used in the example.

- [Authenticate the client](#)
- [Create the thing](#)
- [Get the thing](#)
- [List the things](#)

#### IMPORTANT

Do not combine operations in your examples and their code snippets. For example, do not include both "create database" and "create container" operations in an example. Combining operations can prevent a developer from finding an example when they're scanning your article or using CTRL+F to search it.

## Authenticate the client

This operation (and its H3) is **required** and must be the first code example in the section if your library requires authentication for use.

## Create the thing

Use the [create\\_thing](#) method to create a Thing reference; this method does not make a network call. To persist the Thing in the service, call [Thing.save](#).

```
thing = client.create_thing(id, name)
thing.save()
```

## Get the thing

The [get\\_thing](#) method retrieves a Thing from the service. The `id` parameter is the unique ID of the Thing, not its "name" property.

```
thing = client.get_thing(id)
```

## List the things

Use [list\\_things](#) to get one or more Thing objects from the service. If there are no Things available, a `404` exception is thrown (see [Troubleshooting](#) for details on handling exceptions).

```
things = client.list_things()
```

## Run the code

Provide instruction for running the script or application the reader has created by following your Quickstart. Include the expected output so they can verify successful completion.

## Clean up resources

If you asked the reader to create a resource group, provide instruction for deleting it in this section. If you had them create other resources, provide instruction for deleting those.

## Troubleshooting

This section is **optional**. If you know (or find through verbatims or other avenues) that people commonly run into trouble, help them resolve those issues in this section. For example:

- Describe common errors and exceptions, how to "unpack" them if necessary, and include guidance for graceful handling and recovery.
- Provide information to help developers avoid throttling or other service-enforced errors they might encounter. For example, provide guidance and examples for using retry or connection policies if the library supports it.
- If the client library or a related library supports it, include tips for logging or enabling instrumentation to help them debug their code.

## Next steps

First, briefly summarize the tasks the developer completed in this Quickstart.

Next, use the `[!div class="nextstepaction"]` extension to provide a button that sends the developer to the next logical stop in their journey of using the service in their chosen language. This is typically a **tutorial** if one is

available.

[Process a parallel workload with Python >](#)

# Write a quickstart about deploying Azure services by using ARM templates

5/10/2021 • 6 minutes to read

This article is for Azure services and other products that follow the [Content & Learning content model](#). In addition to standard quickstart requirements, this article describes the high-level process to write a [quickstart](#) focusing on deploying Azure services by using Azure Resource Manager templates (ARM templates).

## Overview

ARM templates are widely used to automate deployments. To ensure customers can deploy Azure services with ARM templates, add a quickstart article to your TOC that shows users how to deploy a common scenario by using an ARM template.

Most MVC-compliant Azure service documentation sites have quickstarts that show how to create the service through the portal, Azure PowerShell, and Azure CLI. Add a fourth one that shows how to do the same task with an ARM template.

If your TOC doesn't have a **Quickstarts** node, create one, and put this quickstart under it.

## Examples

- [Azure Key Vault](#)
- [Azure Load Balancer](#)
- [Azure Cosmos DB](#)
- [Azure Event Grid](#)

## Create an ARM template

This section describes the steps to create an ARM template.

### 1. Define a common scenario.

Identify the infrastructure that users commonly deploy for your service. To simplify learning, try to minimize the number of resources to deploy.

### 2. Review existing ARM templates.

[Azure Quickstart Templates](#) is the template repository managed by the product team. The repository has a web interface that's located at <https://azure.microsoft.com/resources/templates/>. There are likely existing ARM templates that have been created for your Azure service. Review the [links to sample templates in Quickstart repo](#).

Try to find an existing template that matches your common scenario. If one exists, check whether it needs to be updated. Make sure the API versions are the most recent.

To find out the resource provider namespace for Azure services, see [Resource Providers for Azure services](#).

### 3. Develop a template for the common scenario.

If you can't find a template for the common scenario from [Azure Quickstart Templates](#), you need to create

one.

To learn how to create templates, see:

- [ARM template tutorial](#).
- [Quickstart Templates Contribution Guide](#)
- [Quickstart Templates Best Practices Guide](#)

#### 4. Publish the template to Azure Quickstart Templates.

Publish the template before you write the quickstart article. The GitHub repository is [Azure/azure-quickstart-templates](#). The publishing process is similar to MicrosoftDocs.

## Write an ARM template quickstart article for your service

To begin writing your ARM template quickstart article use this [Markdown template](#).

### NOTE

Remove all the comments that are enclosed by `<!-- -->` from the template file before your quickstart article is pushed to GitHub and published.

### TOC

- TOC node under **Quickstarts** that must have a title that includes the words **ARM template**.

Azure Event Grid Documentation

Switch to Event Grid on IoT Edge documentation

> Overview

✓ Quickstarts

  ✗ Storage events

    Azure CLI

    Azure PowerShell

    Portal

**ARM template**

> Custom events

- The TOC node must use **displayName** and have at least **Resource Manager** as a value to enable the TOC search box.

```
7  - name: Quickstarts
8    expanded: true
9    items:
10   - name: Create Cosmos DB resources
11     expanded: true
12     items:
13       - name: Azure portal
14         href: create-cosmosdb-resources-portal.md
15       - name: ARM template
16         displayName: Resource Manager
17         href: quick-create-template.md
18       - name: .NET app - V4 SDK
19         href: create-sql-api-dotnet-v4.md
20       - name: .NET app - V3 SDK
21         href: create-sql-api-dotnet.md
```

### Metadata

Make sure the following metadata is included in your quickstart.

ATTRIBUTE	VALUE
ms.topic	quickstart
ms.custom	subject-armqs

## H1

- The H1 must begin with **Quickstart:** and include the words **ARM template**.

**NOTE**

The style guide team confirmed that for the H1 **ARM template** is preferred because it's more concise and better suited to search results.

## Introduction

The article's introduction must include the following content:

- First paragraph:** Include a sentence that uses Azure Resource Manager template (ARM template) for the first occurrence about the template. For example:

```
This quickstart describes how to use an Azure Resource Manager template (ARM template) to create <service>.
```

**NOTE**

For more information about using **ARM template**, see this [style guide](#) article.

- Second paragraph:** Use the following include file. This include file is a paragraph that consistently introduces ARM concepts before doing a deployment and includes all our desired links to ARM content.

```
[!INCLUDE [About Azure Resource Manager](../../includes/resource-manager-quickstart-introduction.md)]
```

This include file is located in the `/azure-docs-pr/includes/` folder.

- Final paragraph:** Explains that readers who are experienced with ARM templates can continue to the deployment. For information about the button image and how to create the template's URI, see [Deploy the template for Portal](#).

If your environment meets the prerequisites and you're familiar with using ARM templates, select the \*\*Deploy to Azure\*\* button. The template will open in the Azure portal.

```
[![Deploy to Azure](../media/template-deployments/deploy-to-azure.svg)]  
(https://portal.azure.com/#create/Microsoft.Template/uri/<template's URI>)
```

## First H2: Prerequisites

This section must begin with a sentence that includes a link to create a free Azure account. If your service has other prerequisites, list them after the free account sentence.

```
If you don't have an Azure subscription, create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) before you begin.
```

## Second H2: Review the template

- The first sentence must be:

The template used in this quickstart is from [Azure Quickstart Templates] (<https://azure.microsoft.com/resources/templates/<templateName>>).

Use a link to the quickstart gallery that begins with <https://azure.microsoft.com/resources/templates/>. For example, [Azure Key Vault](#).

- After the first sentence, add a JSON code fence that links to the quickstart template. Customers have provided feedback that they prefer to see the whole template. We recommend you include the entire template in your article. If your template is too long to show in the quickstart (more than 250 lines), you can instead add a sentence that says:

The template for this article is too long to show here. To view the template, see [azuredeploy.json] ([link to template's raw output](#))

The syntax for the JSON code fence is:

```
:::code language="json" source="~/quickstart-templates/<TEMPLATE NAME>/azuredeploy.json":::
```

Example:

```
:::code language="json" source="~/quickstart-templates/101-key-vault-create/azuredeploy.json":::
```

The mapping is defined in the `openpublishing.publish.config` file.

- After the JSON code fence, a list of each `resourceType` from the JSON must exist with a link to the template reference starting with `/azure/templates`. List the `resourceType` links in the same order as in the template.

Two Azure resources are defined in the template:

- [Microsoft.KeyVault/vaults](#): create an Azure key vault.
- [Microsoft.KeyVault/vaults/secrets](#): create an key vault secret.

The URL usually appears as, for example,

<https://docs.microsoft.com/azure/templates/microsoft.network/loadbalancers> for `loadbalancer` of `Microsoft.Network`.

### NOTE

Remove the API version from the URL so that the URL redirects to the latest version.

- (Optional) List additional quickstart templates. For example:

[Azure Quickstart Templates by resource types] (<https://azure.microsoft.com/resources/templates/?resourceType=Microsoft.KeyVault&pageNumber=1&sort=Popular>).

### NOTE

Notice the `resourceType` and `sort` elements in the URL.

## Third H2: Deploy the template

One of the following options must be included:

- **CLI:** An Azure CLI interactive code fence must contain `az deployment group create`. For example:

```
read -p "Enter a project name that is used for generating resource names:" projectName &&
read -p "Enter the location (i.e. centralus):" location &&
templateUri="https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-
account-create/azuredeploy.json" &&
resourceGroupName="${projectName}rg" &&
az group create --name $resourceGroupName --location "$location" &&
az deployment group create --resource-group $resourceGroupName --template-uri $templateUri &&
echo "Press [ENTER] to continue ..." &&
read
```

### NOTE

`az deployment group create` requires Azure CLI version 2.6 or later. To display the version type `az --version`. For more information, see the [documentation](#).

- **PowerShell:** An Azure PowerShell interactive code fence must contain `New-AzResourceGroupDeployment`.

For example:

```
$projectName = Read-Host -Prompt "Enter a project name that is used for generating resource names"
.setLocation = Read-Host -Prompt "Enter the location (i.e. centralus)"
$templateUri = "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-
storage-account-create/azuredeploy.json"

$resourceGroupName = "${projectName}rg"

New-AzResourceGroup -Name $resourceGroupName -Location "$location"
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateUri $templateUri

Read-Host -Prompt "Press [ENTER] to continue ..."
```

- **Portal:** Use a button with the description **Deploy to Azure** and the shared image `../media/template-deployments/deploy-to-azure.svg`. Please use this shared image, instead of an image in one of your folders, so it can be easily updated in the future. The template link starts with

`https://portal.azure.com/#create/Microsoft.Template/uri/`:

```
[!Deploy to Azure](../media/template-deployments/deploy-to-azure.svg)
(https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2Fazure-quickstart-templates%2Fmaster%2F101-key-vault-create%2Fazuredeploy.json)
```

- For more information about this deployment option and how to create the template's URI, see [Use a deployment button to deploy templates from GitHub repository](#).
- The shared button image is in [GitHub](#).

## Fourth H2: Review deployed resources

This heading must be titled **Review deployed resources** or **Validate the deployment**. Include at least one method that displays the deployed resources. Use a portal screenshot of the resources, or interactive code fences for Azure CLI (`azurecli-interactive`) or Azure PowerShell (`azurepowershell-interactive`).

## Fifth H2: Clean up resources

The **Clean up resources** section includes a paragraph that explains how to delete unneeded resources. Include

at least one method that shows how to clean up resources. Use a portal screenshot, or interactive code fences for Azure CLI (`azurecli-interactive`) or Azure PowerShell (`azurepowershell-interactive`).

See [Examples](#).

## Sixth H2: Next steps

Make the next steps similar to other [quickstarts](#) and use a blue button to link to the next article for your service. Or direct readers to [Tutorial: Create and deploy your first ARM template](#) to follow the process of creating a template.

```
> [<div class="nextstepaction">]
> [Tutorial: Create and deploy your first ARM template](/azure/azure-resource-manager/templates/template-tutorial-create-first-template)
```

To include additional links for more information about the service, it's acceptable to use a paragraph and bullet points.

# Write tutorials

5/28/2021 • 8 minutes to read

Tutorials lead a user through creating a proof of concept. The goal is to help them understand how to do the task later in a way that is specific for their environment. Therefore, we present only one way to do each part of the task. They are typically longer and more complex than quickstarts. You create tutorials from the list of top user tasks identified in milestone one and they focus on the **single** best procedure. A tutorial usually aligns to one or two customer tasks. You need at least one tutorial in the Tutorials node of the TOC.

Tutorials are 100 – 200 level for an audience new to the service, product, or scenario. Ideally, your tutorials should build on each other using a common theme with the result being a completed customer scenario. Unlike how-to content, they are not open-ended walkthroughs of a feature.

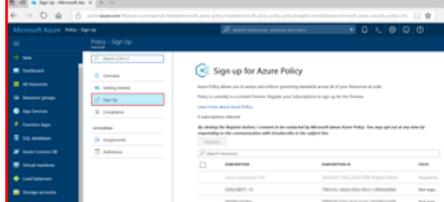
Use [this template](#) when writing a new tutorial article.

## Tutorial checklist

## Tutorial

<Service name>  
Documentation

- > Overview
- > Quickstarts
- ✓ Tutorials
  - 1 <customer task>
  - <customer task>
  - <customer task>
  - <customer task>
  - <customer task>
- > Samples
- > Concepts
- > How-to guides
- > Reference
- > Resources

2 H1  
In this tutorial, you learn how to:  
3 ✓  
✓  
✓  
4 If you don't have an Azure subscription, create a [free account](#) before you begin.  
  
5 H2  
6 Clean up resources  
7 Next steps  
Advance to the next article to learn about...

NUMBER	REQUIREMENT
1	Titles in the TOC don't wrap to a second line.
2	Start your H1 with "Tutorial:"
3	Before your first H2, use the <i>green</i> checkmark format for the bullets that outline what you'll cover in the tutorial.
4	Before the first H2, provide a link to a free account if one exists.
5	If users access your product/service via a web browser, include the browser in the first screenshot in a tutorial.
6	Clean up resources (H2) should come just before Next steps (H2)
7	Next steps (H2) section is required. A single link in the blue box should direct the customer to the next tutorial.

The checklist above covers key guidelines. The sections below provide more information to help make sure your tutorial meets additional content quality criteria.

## TOCs

Each TOC needs a Tutorials node. Don't come up with other TOC nodes on your own.

Tutorial titles in the TOC don't contain the word "Tutorial." If the tutorials comprise a series, number them in the TOC. Unless it is unavoidable, make sure your titles don't wrap to a second line.

If there is no Quickstarts section in the TOC, expand the Tutorials section by default.

## Templates

Use this [standard tutorial template](#) unless there is a more product/service/technology appropriate template in the repository-specific section of the guide.

## File name

If you are repurposing an existing article, consider keeping the existing file name unless it's misleading. That way you avoid redirects and keep the data about the article.

## Metadata

Make sure the following metadata is included in your tutorial.

ATTRIBUTE	VALUE
ms.topic	tutorial

### Title attribute

Include the word "tutorial" in the title attribute.

If the tutorial is part of a numbered series, don't include the number in the Title attribute.

### Description attribute

Include the word "tutorial" in the Description attribute. "In this tutorial, you..."

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This statement provides a record of the intent of this tutorial for future contributors.

## H1 (Headline)

Start your H1 with "Tutorial:"

Make the first word following "Tutorial:" a verb.

If the tutorial is part of a numbered series, don't include the number in the H1

Don't add "Tutorial:" to the H1 (or page title) of any article not in the Tutorials node of the TOC.

## Introduction

Lead with a light intro that describes, in customer-friendly language, what they'll learn, or do, or accomplish. Answer the fundamental "why would I want to do this?" question. Avoid starting the tutorial with a bunch of notes or caveats.

Include a sentence that says, "In this tutorial you will do X..."

Before your first H2, use the *green checkmark* format for the bullets that outline what you'll cover in the tutorial.

The markdown code for the green checkmarks looks like this:

```
> [&gt;div class="checklist"]
> * Create and connect to a VM
> * Select and use VM images
> * View and use specific VM sizes
> * Resize a VM
> * View and understand VM state
```

Don't include items in this list that are covered in the **Prerequisites** section of a tutorial.

### Free trial account information

Because tutorials are intended to help new customers use the product or service to complete a top task, include a link to a free trial before the first H2, if one exists.

Some examples of trial account info.

- **Azure**

```
If you don't have an Azure subscription, create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) before you begin.
```

For Azure, don't provide instructions or links to the "try it now" experiences. The "try it now" experiences are for marketing content; documentation should link to the free trials that can be easily converted to paid accounts.

- **PowerApps**

If you're not signed up for PowerApps, [sign up for free](<https://web.powerapps.com/signup?redirect=marketing&email=>) before you begin.

- **Power BI**

If you're not signed up for Power BI, [sign up for a free trial]([https://app.powerbi.com/signupredirect?pbi\\_source=web](https://app.powerbi.com/signupredirect?pbi_source=web)) before you begin.

## Includes

For tutorials that use the CLI and PowerShell, test the procedure using Azure Cloud Shell. If you can run all the CLI or PowerShell steps in Cloud Shell, use the following /include file as a step after the prerequisites.

- For the Azure CLI

```
[!INCLUDE [cloud-shell-try-it.md](../../../../includes/cloud-shell-try-it.md)]
```

If all commands don't run in Cloud Shell, do not use the /include files. Instead, instruct the user to install the CLI or PowerShell locally. Include the following text in an H2 Prerequisites section. Replace the <version #> placeholder with specific version requirements.

- For the Azure CLI

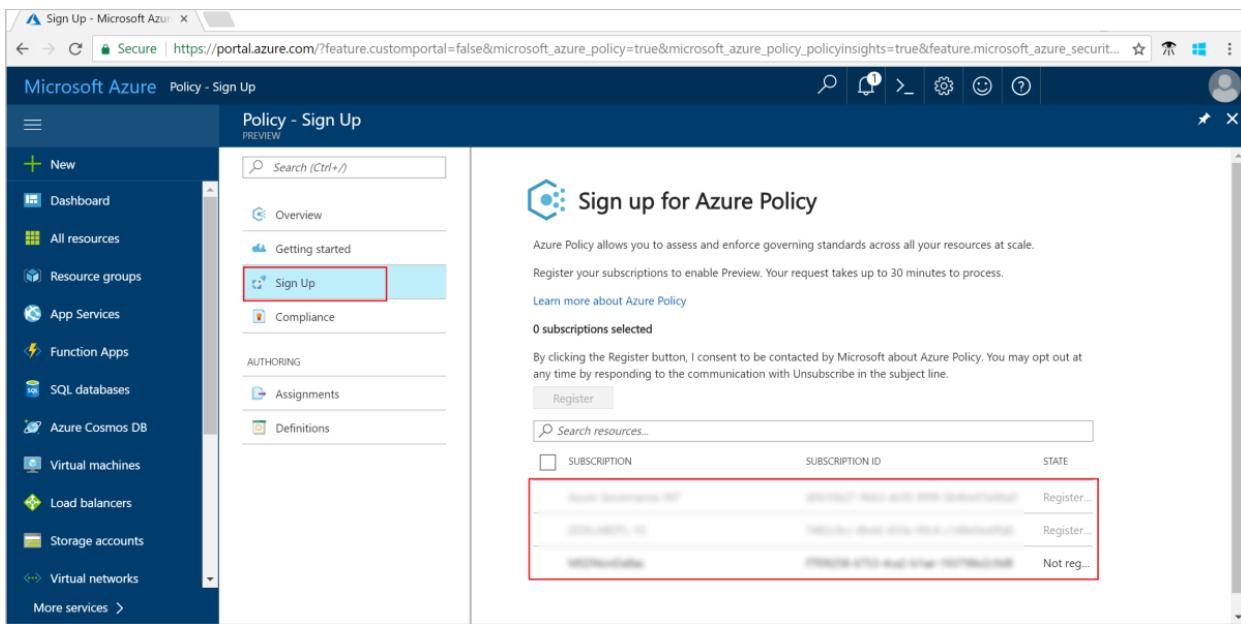
This tutorial requires that you run the Azure CLI locally. You must have the Azure CLI version <version #> (for example 2.0.4) or later installed. Run `az --version` to find the version. If you need to install or upgrade the CLI, see [Install Azure CLI 2.0](#).

- For Azure PowerShell

This tutorial requires that you run PowerShell locally. You must have Azure PowerShell module version <version #> (for example 5.1.1) or later installed. Run `Get-Module -ListAvailable Azurerm` to find the version. If you need to upgrade, see [Install Azure PowerShell module](#). After you verify the PowerShell version, run `Login-AzurermAccount` to create a connection with Azure.

## Screenshots

If users access your product/service via a web browser the first screenshot should always include the **full browser window in Chrome or Safari**. This perspective is to show users that the portal is browser-based - OS and browser agnostic.



Make sure that additional screenshots offer value for the customer (explaining complex UI, providing context, validating they've taken the correct steps or are in the right place, etc.). Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article.

See additional [art guidelines for screenshots](#).

## H2s and H3s

Don't number H2 and H3 headings. Tutorials are procedure-based and customers understand that the sections follow in sequence.

Follow the H2 headings with one or two transitional sentences explaining why the steps need to be performed or how that section contributes to the whole.

## Prerequisites

Make **Prerequisites** your first H2 in a tutorial. If there's something a customer needs to address before they start (for example, creating a VM) it's OK to link to that content before they begin.

## Linking from the procedure

Tutorials are intended to guide the customer through an end-to-end procedure. Don't link off to other content in the procedural part of the tutorial. For example, if the customer needs to set permissions, include the permissions they need to set, and the specific settings in the tutorial procedure. Don't send the customer to another article to read about it.

In a break from tradition, do not link to reference topics in the procedural part of the tutorial when using cmdlets or code. Provide customers what they need to know in the tutorial to successfully complete the tutorial.

## Bulleted and numbered lists

Minimize bullets and numbered lists in tutorials.

## Alerts

Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.

If you need to use alerts, limit them to one or two per article. Multiple notes should never be next to each other in an article.

## Code blocks

Code requires specific formatting. Here are a few useful examples of commonly used code blocks. Make sure to use the interactive functionality where possible.

Here is an example of a code block for Java:

```
cluster = Cluster.build(new File("src/remote.yaml")).create();
...
client = cluster.connect();
```

or an interactive code block for Azure CLI:

```
az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter --admin-username
azureuser --admin-password myPassword12
```

or an interactive code block for Azure PowerShell:

```
New-AzureRmContainerGroup -ResourceGroupName myResourceGroup -Name mycontainer -Image
mcr.microsoft.com/windows/servercore/iis:nanoserver -OsType Windows -IpAddressType Public
```

## Clean up resources

If you're creating a standalone tutorial that isn't part of a series, include guidance on how to clean up or delete resources that the customer created in the tutorial. Create a **Clean up resources** H2, and use text similar to this example:

### Clean up resources

When no longer needed, delete the resource group, virtual machine, and all related resources. To do so, select the resource group for the VM and click **Delete**.

## Next steps

Tutorials should always have a **Next steps** H2 that points to the next logical tutorial in a series (one link only here). If there are no other tutorials, point to some other thing the customer can do with the service. Use the blue box format for tutorials. You can shorten the H1 in the boxes if the original one doesn't fit.

### Next steps

In this tutorial, you learned the basic steps to extend the storage capacity of a Windows server by using Azure File Sync. For a more thorough look at planning for an Azure File Sync deployment, see:

[Plan for Azure File Sync deployment](#)

The markdown code for the blue box looks like this:

```
> [&gt;!div class="nextstepaction"]
> [Azure Windows virtual machine tutorials](./tutorial-manage-vm.md)
```

Do not link to troubleshooting content or FAQs. Including this content implies the customer is going to run into

known issues. Either fix these issues beforehand or create a different path to success for users that avoid these problems.

# Write a concept article

5/28/2021 • 5 minutes to read

Concept articles are primarily distinguished by what they aren't: they aren't procedural articles. They don't have specific end states (other than conveying an underlying idea) and don't have concrete, sequential actions for the user to do. One clear sign of a procedural article would be the use of a numbered list; with rare exception, numbered lists shouldn't appear in concept articles.

## What's a "concept"?

If an idea or principle is common or useful among various tasks, it might be an appropriate topic for a concept article. Concepts aren't necessarily complex or complicated, and if explained adeptly, should be easily grasped by the reader regardless. However, be sure to respect the reader and meet them at their level. Readers who examine intricate ideas likely already have foreknowledge and experience in the subject. In such cases, the simplest explanation for an advanced concept may still be advanced.

There isn't a hard rule for the scope of a concept. Concepts can be iterative and recursive and build upon each other to form entire domains of study or disciplines. An example of nesting is how the concept of metabolism is part of organic chemistry, which is itself a branch of biochemistry, and so on. As an automotive example, a user doesn't need to know the principles of "torque" (rotational force) when doing a "tire change", they just need to know that the lug nuts on the wheel should be "set" to a specific "torque" with a "torque wrench."

Likewise with computing technology, to configure networking settings for a host, a user doesn't need a detailed explanation of how resource requests are technically served, or how packet encapsulation works. Such an understanding could help the user accelerate or customize the process, but it isn't necessary for the successful completion of a task.

## Why would I use a concept article?

There are several instances that may call for the writing of a concept article, including:

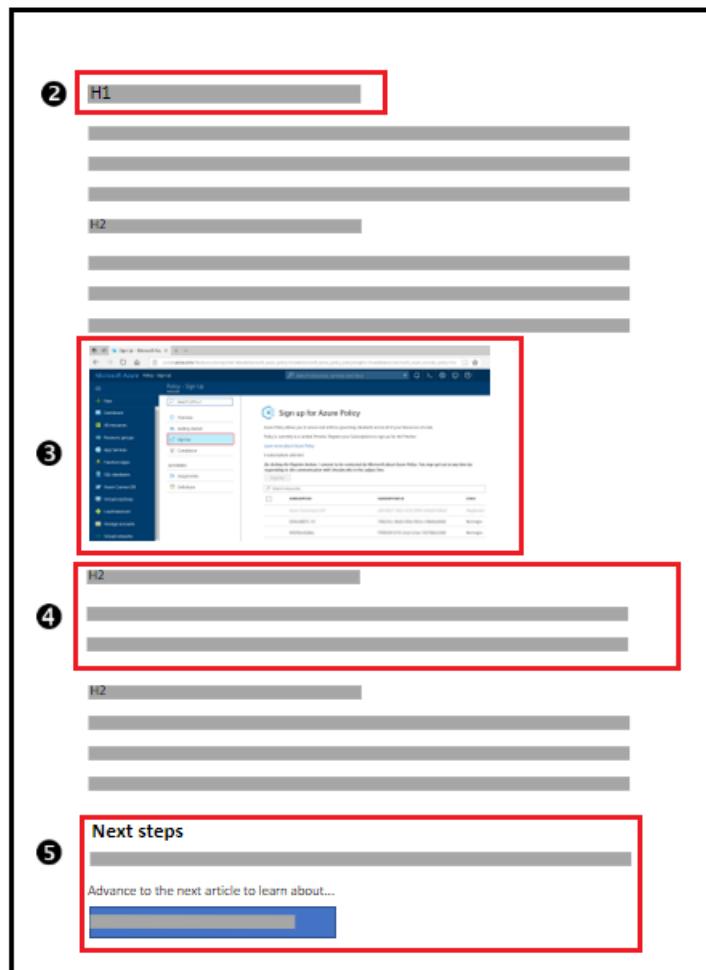
- If there's a 'new' idea that's central to the service or product, that idea must be articulated so that customers can understand the value of the service or product as it relates to their circumstances. A good recent example is the concept of "containerization" or the concept of "scalability".
- If there's optional information or explanations that are common to several tutorials or [how-to guides](#) these can be consolidated and single-sourced in a full-bodied concept article for you to reference.
- If the service or product is extensible, advanced users might modify (mod) it to better suit their application. It's better that advanced users fully understand the reasoning behind the design choices and everything else "under the hood" so that their variants are more robust, thereby improving their experience.

Use [this template](#) when writing a new concept article.

## Concept article checklist

# Concept

<Service name>
Documentation
> Overview
> Quickstarts
> Tutorials
> Samples
✓ Concepts
① <Concept>
<Concept>
> How-to guides
> Reference
> Resources



NUMBER	REQUIREMENT
1	Titles in the TOC don't wrap to a second line.
2	Start your H1 with a noun.
3	When appropriate, include diagrams, tables, or imagery to reinforce a concept.
4	Include at least two H2 sections not including prerequisites and next steps.
5	Next steps (H2) section is required. Provide at least one next step and no more than three.

The checklist above covers key guidelines. The sections below provide more information to help ensure your article meets more content quality criteria.

## TOCs

Start the TOC title with a noun. A concept is an idea that undergirds an action, not the action itself.

Unless it's unavoidable, make sure your titles don't wrap to a second line.

## File name

If you are reusing an existing article, consider keeping the existing file name unless it's misleading. That way you avoid redirects and keep the data about the article.

## Metadata

Make sure the following metadata is included in your how-to guide.

ATTRIBUTE	VALUE
ms.topic	conceptual

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This statement provides a record of the intent of the concept for future contributors.

## H1 (Headline)

Pick an H1 that clearly conveys what the content's about. As concepts deal with ideas or principles and *not* actions or steps, this headline wouldn't normally be a verb or insinuate a concrete task.

EXAMPLE	TEXT
X Not ideal	Share dashboards with Azure role-based access control
O Better	Role-based access control for Azure dashboards ( <i>OR</i> ) Azure dashboard rose-based sharing

## Introduction

Lead with a light intro that describes, in customer-friendly language, what they'll learn. Answer the fundamental "why would I want to learn this knowledge?" question. Avoid starting the article with a bunch of notes or caveats.

For definitive concepts, it's better to lead with a sentence in the form, "X is a (type of) Y that does Z"

EXAMPLE	TEXT
X Not ideal	You should apply the new technology of containerization to make all your applications work on all your systems.
O Better	Containerization is a form of application encapsulation that bundles all related program files and libraries into a single executable package so an application can run on any host operating system.

## Screenshots

Make sure that extra screenshots offer value for the customer (explaining complex UI, providing context, validating they've taken the correct steps or are in the right place, and so on). Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article.

See more [art guidelines for screenshots](#).

## H2s and H3s

Don't number H2 and H3 headings.

Follow the H2 headings with one or two transitional sentences explaining how that section contributes to the whole.

## Prerequisites

If there are prerequisites (other concepts) for the idea covered in a concept article, make **Prerequisites** your first H2 in the guide.

## Bulleted lists

Minimize bullets in concept articles.

## Alerts

Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.

If you need to use alerts, limit them to one or two per article. Multiple notes should never be next to each other in an article.

## Code blocks

Code requires specific formatting. Here are a few useful examples of commonly used code blocks. Make sure to use the interactive functionality where possible.

Here's an example of a code block for Java:

```
cluster = Cluster.build(new File("src/remote.yaml")).create();
...
client = cluster.connect();
```

or an interactive code block for Azure CLI:

```
az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter --admin-username
azureuser --admin-password myPassword12
```

or an interactive code block for Azure PowerShell:

```
New-AzureRmContainerGroup -ResourceGroupName myResourceGroup -Name mycontainer -Image
mcr.microsoft.com/windows/servercore/iis:nanoserver -OsType Windows -IpAddressType Public
```

## Next steps

Concept articles should always have a **Next steps** H2 that points to the next logical article (one to three links only here). These links can be other concept articles or tutorials are relevant to the concept at hand.

Don't link to troubleshooting content or FAQs. Including this content implies the customer is going to run into known issues. Either fix these issues beforehand or create a different path to success for users that avoid these problems.

# Write how-to guides

6/10/2021 • 3 minutes to read

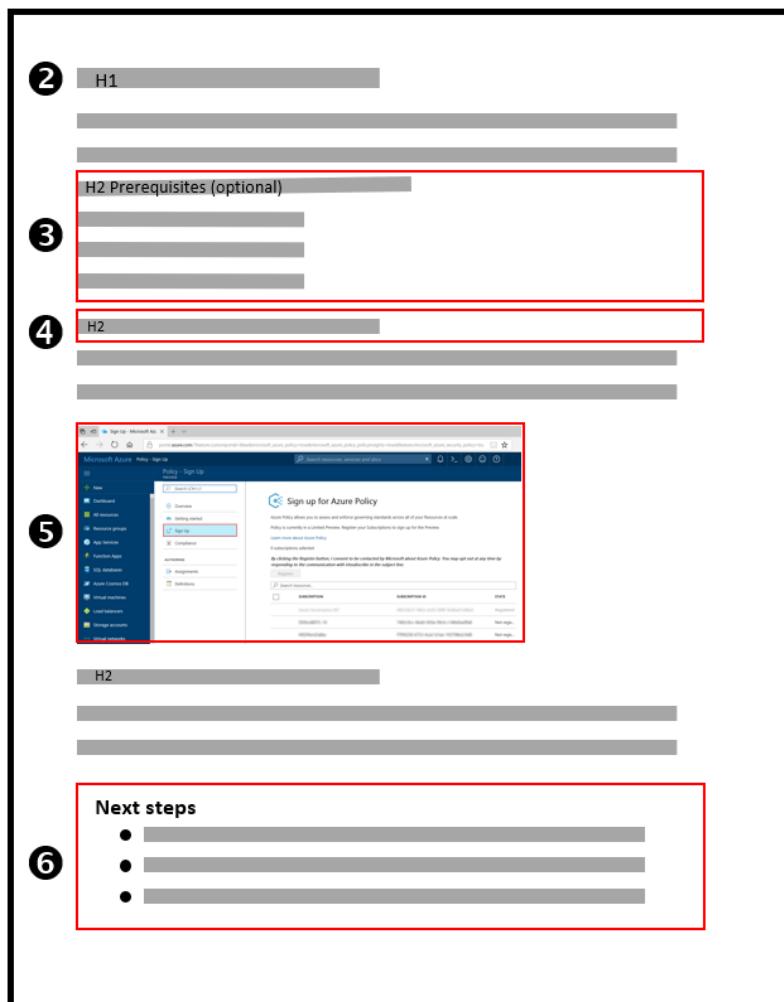
How-to guides are primarily procedural and help customers complete tasks in their own environment. They differ from tutorials in that they can include optional information, explanations, and information to help inform decisions.

Use [this template](#) when writing a new how-to guide.

## How-to guide checklist

# How-to guides

- <Azure service name>  
Documentation
- > Overview
  - > Quickstarts
  - > Tutorials
  - > Samples
  - > Concepts
  - > How-to guides
    - 1 > Customer task
    - > Customer task
    - > Customer task
- Reference  
Resources



NUMBER	REQUIREMENT
1	Titles in the TOC don't wrap to a second line.
2	Start your H1 with a verb
3	If you need prerequisites, make them your first H2 in a how-to guide.

NUMBER	REQUIREMENT
4	Include at least two H2 sections not including prerequisites and next steps.
5	If users access your product/service via a web browser, include the browser in the first screenshot in a how-to guide.
6	Next steps (H2) section is required. Provide at least one next step and no more than three.

The checklist above covers key guidelines. The sections below provide more information to help make sure your how-to guide meets additional content quality criteria.

## TOCs

While not required, most TOCs have a **How-to guides** node. Don't come up with other TOC nodes on your own.

Start the TOC title with a verb.

Unless it is unavoidable, make sure your titles don't wrap to a second line.

## File name

If you are repurposing an existing article, consider keeping the existing file name unless it's misleading. That way you avoid redirects and keep the data about the article.

## Metadata

Make sure the following metadata is included in your how-to guide.

ATTRIBUTE	VALUE
ms.topic	how-to

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This statement provides a record of the intent of this how-to guide for future contributors.

## H1 (Headline)

Start your H1 with a verb

Pick an H1 that clearly conveys what the content is about.

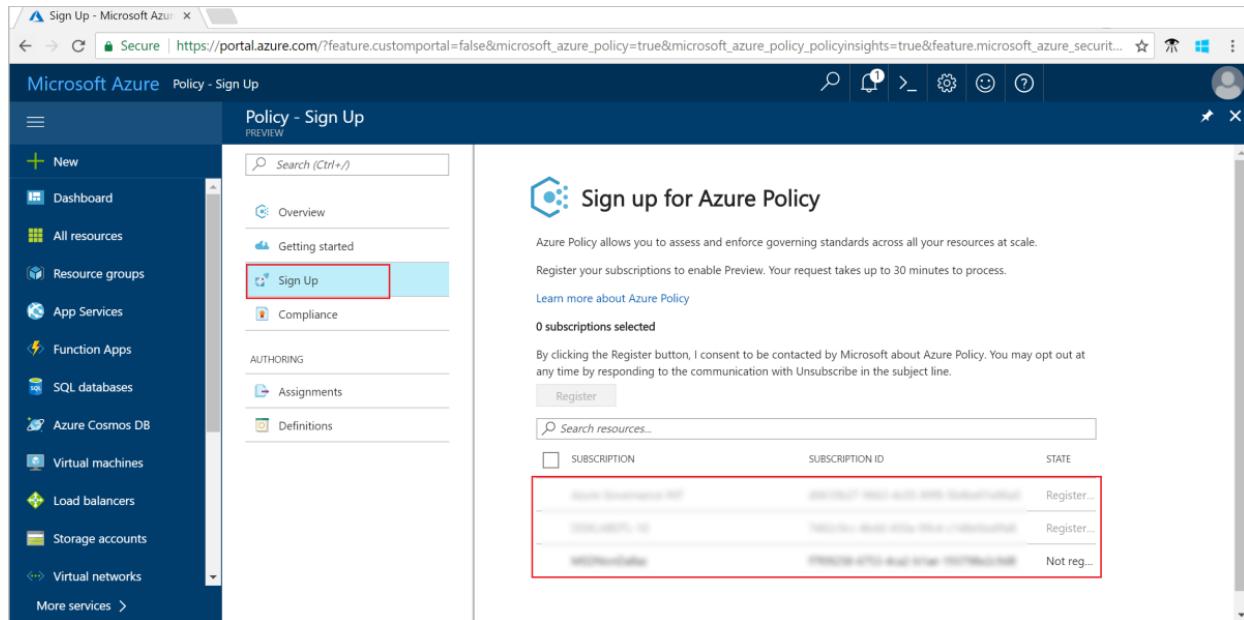
## Introduction

Lead with a light intro that describes, in customer-friendly language, what they'll learn, or do, or accomplish. Answer the fundamental "why would I want to do this?" question. Avoid starting the article with a bunch of notes or caveats.

Include a sentence that says, "In this how-to guide you will do X..."

## Screenshots

If users access your product/service via a web browser the first screenshot should always include the **full browser window in Chrome or Safari**. This perspective is to show users that the portal is browser-based - OS and browser agnostic.



Make sure that additional screenshots offer value for the customer (explaining complex UI, providing context, validating they've taken the correct steps or are in the right place, etc.). Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article.

See additional [art guidelines for screenshots](#).

## H2s and H3s

Don't number H2 and H3 headings. How-to guides are procedure-based and customers understand that the sections follow in sequence.

Follow the H2 headings with one or two transitional sentences explaining why the steps need to be performed or how that section contributes to the whole.

## Prerequisites

If there are prerequisites for the task covered in a how-to guide, make **Prerequisites** your first H2 in the guide.

## Bulleted lists

Minimize bullets in how-to guides.

## Alerts

Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.

If you need to use alerts, limit them to one or two per article. Multiple notes should never be next to each other in an article.

## Code blocks

Code requires specific formatting. Here are a few useful examples of commonly used code blocks. Make sure to use the interactive functionality where possible.

Here is an example of a code block for Java:

```
cluster = Cluster.build(new File("src/remote.yaml")).create();
...
client = cluster.connect();
```

or an interactive code block for Azure CLI:

```
az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter --admin-username
azureuser --admin-password myPassword12
```

or an interactive code block for Azure PowerShell:

```
New-AzureRmContainerGroup -ResourceGroupName myResourceGroup -Name mycontainer -Image
mcr.microsoft.com/windows/servercore/iis:nanoserver -OsType Windows -IpAddressType Public
```

## Next steps

How-to guides should always have a **Next steps** H2 that points to the next logical article (one to three links only here).

Do not link to troubleshooting content or FAQs. Including this content implies the customer is going to run into known issues. Either fix these issues beforehand or create a different path to success for users that avoid these problems.

# FAQ best practices

4/30/2021 • 3 minutes to read

## NOTE



THIS DOCUMENT IS IN REVIEW AND IS NOT YET SUPPORTED IN THE CONTENT STANDARDS FOR [DOCS.MICROSOFT.COM](#). WE ENCOURAGE YOU TO USE THE GUIDANCE AND PROVIDE FEEDBACK [IN OUR TWO-QUESTION SURVEY](#).

The frequently asked questions (FAQ) format structures information in a question-driven manner. If someone is new to your products or services, what might they have a question about?

## Should you write a FAQ?

If you have to, such as if you're a part of an organization that prefers that your documents include an FAQ, or if you want to customize search engine results, then you'll be writing a FAQ. But modern authoring technology often provides better ways to answer a user's questions, and standard article types (for example, a [tutorial](#)) generally cover the same content a that FAQ would, but more efficiently and can still be easily found via search.

Therefore, if you write a FAQ, *don't duplicate content*. The ideal FAQ keeps entries as short as is necessary and instead points off to the "meaty" articles where the subject is covered in more detail. The [Windows Virtual Machines FAQ](#) is an excellent example of providing a direct answer and then referencing an appropriate resource for further details.

**Can I resize the OS disk?**

Yes. For instructions, see [How to expand the OS drive of a Virtual Machine in an Azure Resource Group](#).

## How should I organize an FAQ?

Start with the most common or significant customer cases. Chunk similar questions and answers together when appropriate (for example, if a process is similar in all but one step in various operating systems).

**Why does my engine barely start / not start, or die after starting?**

*If your vehicle is a conventional gasoline engine car, the issue could be caused by your battery; spark plugs; a lack of fuel caused by an empty or leaky tank, fuel pump, or injectors; starter motor; cylinder pressure; or, other electrical or module type issue, such as valve/gear timing error detection.*

Sometimes, things are so obvious, you wouldn't think they should even be noted. It seems silly, for instance, that a driver wouldn't know that you need to fill a car with gas, but a novice can overlook such a point.

## How specific should questions in an FAQ be?

Don't be unnecessarily specific unless you are creating your FAQ in a [specific YML format](#). Overly specific FAQ questions create a high volume of content with a low relative impact, but this cost is reduced if the user doesn't need to sort through the other options.

**Are you open at night?**

*No, we are not open for dinner service.*

**Can I order take out for dinner?**

*We are not open for dinner service and do not have a dinner menu. Take out is available for breakfast or lunch.*

**Are you open during the day?**

*Yes, we are open from 9AM-4PM.*

**What time is the kitchen open?**

*The kitchen serves food from 9AM-4PM, Monday to Friday.*

However, *don't* generalize your content if you know your customer is referring to a specific issue or circumstance.

**What is "This hosts requires Upgrade DR-86-RS" message and how do I stop it from popping up?**

*This upgrade notification appears if a machine has an Atlantic series processor. A mandatory security vulnerability patch must be downloaded and applied to clear the warning. (*

*Main > System and Settings > Updates > Update now )*

In this case, the asker has no intent of understanding the vulnerability - and they don't need to, provided they have the right configuration. If you can't provide a clear answer, provide an educated guess.

## What is a structured FAQ?

A "structured FAQ" refers to [a specific YML format](#) optimized for search results pages. The [Azure Active Directory device management FAQ](#) is an example of a structured FAQ. A YML FAQ can be an effective form of documentation for one-off corner-cases or "gotchas." *Bear in mind: regular (unformatted) content in well-designed, popular pages appears in search results by default.*

### TIP

When decided the best keywords for your content, remember that customers want an answer, but they are *searching* for the answer within the language of their problem.

# Create FAQ content

3/5/2021 • 3 minutes to read

When you have frequently asked questions (FAQ) and answers about products, features, and services, you can use a structured YAML template to present the content. The purpose of using a structured YAML template is to align with a specific schema that Google has defined for FAQ pages. That schema provides better crawlability, ranking, and relevance in the search engine.

## Benefits of using an FAQ template

If FAQ pages are created with the Google-defined schema, the search engine will crawl, index, and rank the content for all inquiry-related search queries. Google presents the FAQ schema-based content in a question-and-answer format on a search engine results page (SERP). Because questions appear directly on a Google SERP with expandable answers, users can instantly get the information they're looking for and gives a lot more real estate on search engine result page.

When the FAQ content is correctly marked up with Google defined schema, these pages could be eligible for Google rich snippets in the search results and may appear in Google Assistant searches.

Here's an example of how FAQ content appears on a Google SERP with questions:

The screenshot shows a Google search results page for the query "bitlocker requirements FAQ". The top result is from docs.microsoft.com, titled "BitLocker overview and requirements FAQ (Windows 10 ...)". Below the title is a snippet of text: "Feb 28, 2019 – Two partitions are required to run BitLocker because pre-startup authentication and system integrity verification must occur on a separate partition ...". Underneath this, there is a list of four expandable FAQ questions with green arrows pointing to the right next to each question:

- How does BitLocker work?
- Does BitLocker support multifactor authentication?
- What are the BitLocker hardware and software requirements?
- What credentials are required to use BitLocker?

Here's an example of how FAQ content appears on a Google Featured Snippet on top position.



microsoft edge frequently asked questions

X |

All News Images Videos Shopping More Settings Tools

About 470,000,000 results (0.63 seconds)

### Frequently Asked Questions (FAQ) for IT Pros

- How can I get the next major version of **Microsoft Edge**, based on Chromium? ...
- What's the difference between **Microsoft Edge** and Internet Explorer 11? ...
- Does **Microsoft Edge** work with Enterprise Mode? ...
- How do I customize **Microsoft Edge** and related settings for my organization? ...
- Is Adobe Flash supported in **Microsoft Edge**?

[More items...](#) • Dec 16, 2020

docs.microsoft.com › Docs › Microsoft Edge deployment

[Microsoft Edge - Frequently Asked Questions \(FAQ\) for IT Pros ...](#)[>About featured snippets](#) [Feedback](#)

## Structure of an FAQ page

The FAQ template is YAML based, though it allows Markdown within the answer section and the section for additional content.

PROPERTY	TYPE	REQUIRED	DESCRIPTION
metadata	object	yes	
title	string	yes	Title text that Google crawls for ranking
description	string	yes	Metadata description set for SEO
title	string	yes	H1 at the top of the page
summary	Markdown	no	Summary after the H1 and before question-and-answer content
sections	array	yes	List of sections of content
name	string	yes	Title of a section, used as an H2
questions	array	yes	List of questions in this section
question	string	yes	Question text
answer	Markdown	yes	Answer text

PROPERTY	TYPE	REQUIRED	DESCRIPTION
additionalContent	Markdown	no	Optional additional content for the bottom of the page

#### NOTE

If you have a short list of questions and you don't need multiple sections, use a single section. The heading will be ignored, and each question will be rendered as an H2 for **In this article** navigation.

## Examples

Here's an example of FAQ content with two sections:

```
###YamlMime:FAQ
metadata:
  title:MicrosoftcommercialmarketplacepublisherFAQ
  description:"Getanswers to common questions about Azure Marketplace and Microsoft AppSource."
  title:CommonquestionsabouttheMicrosoftcommercialmarketplace
  summary:This article answers commonly asked questions about the commercial marketplace.

sections:
  -name:General
    questions:
      -question:What is the Microsoft commercial marketplace?
      answer:|
        The commercial marketplace connects business and government agencies with solutions built by our partners. Our partners create and manage offers in Partner Center, and customers can discover and buy solutions...
        To learn more, go to the [Microsoft commercial marketplace hub]
        (https://partner.microsoft.com/asset/collection/commercial-marketplace#/).

      -question:Where can I learn more about the Microsoft Admin Center?
      answer:For information about the Microsoft Admin Center, see [Manage third-party app subscriptions for your organization]
        (https://microsoft-365.commerce/manage-saas-apps).

  -name:Resources
    questions:
      -question:Where can I find more information about the commercial marketplace?
      answer:|
        Here are some resources to get you started:
        - [What is the Microsoft commercial marketplace?](overview.md)
        - [Microsoft commercial marketplace partner hub]
        (https://partner.microsoft.com/asset/collection/commercial-marketplace#/)

  additionalContent: |
    ##InConclusion
    Here's some optional text that can be placed at the end of the document.
```

For complete examples, see:

- [Example YAML with multiple sections](#)
- [Example YAML with single section](#)

## Steps to convert existing FAQ content into the new template

To convert an FAQ page that's currently written as a conceptual Markdown file:

1. Use the `title` and `description` fields in the metadata table for the `metadata.title` and `metadata.description` properties.
2. Copy and paste all the existing metadata from current page to YAML file.
3. Use the H1 content (content starting with a single `#`) as `title`.
4. Use any introductory content as `summary`.
5. To use any formatting in the summary, like alerts, prefix with `|`.
6. Copy each answer and paste it into the new file. Multiline answers or answers with formatting probably need to be prefixed with `|` and a new line, as shown in the [example sections](#).

Live examples converted from MD to FAQ YAML:

1. <https://docs.microsoft.com/windows/security/information-protection/bitlocker/bitlocker-overview-and-requirements-faq>
2. <https://docs.microsoft.com/microsoftteams/faq-journey>

### FAQ markdown to yml conversion utility

The engineering team has built a utility to convert MD FAQ pages into YAML template which can be very useful and handy for easy conversion. You can find the utility at [MD to FAQ conversion tool](#)

### Additional Information

- You don't need to change the file name or folder for the new template and you can keep the same file name to avoid redirection but new files must have a .yml extension.
- Make sure to rename the existing MD file otherwise OPS build will produce error if it finds the two files with same name even though one is MD and second one YAML. Once the new .yml is published and everything looks good delete the old MD faq page from the repo.
- Update the TOC.yml to point to new .yml file instead of old MD file.
- Update the other links within the repo including the landing page if FAQ MD page is mentioned on the landing page.

# Zone pivots (preview)

6/4/2021 • 10 minutes to read

## NOTE

Updates to any `zone-pivot-groups.yml` configuration file should be peer reviewed. Please keep unused groups out of the file.

Zone pivots enhance navigation, allowing readers to toggle the content in an article using grouped tabs -- all without reloading the page. From a writer's perspective, zone pivots allow you to make a section (or "zone") in an article and specify when it is rendered for the reader.

This article covers when to use zone pivots, best practices for implementation, and instructions for adding them to your articles. The zone pivots feature is still considered to be *in preview*.

## Zone pivots vs. tabbed conceptual

The platform exposes two features that allow writers to add tab-based content selection to their documentation: **zone pivots** and **tabbed conceptual**. Both features:

- Allow for consolidation of content on a single page
- Allow readers to toggle content based on their needs or use case.

However, there are some key differences:

ZONE PIVOT	TABBED CONCEPTUAL
When using zone pivots, the user can make a selection at the top of the document. Once this selection is made, any content that lives within zone pivot code blocks are changed to match that selection.	When using tabbed conceptual, the user can make a selection anywhere within the document where tabs are used. When a selection is made, all tabs within a document will switch to match the selection.
After a user makes a selection using zone pivots, the context is maintained as they navigate the developer website. This means that if a user selects "Python" as a language in the zone pivot, when they navigate to another page that uses this zone pivot group "Python" is automatically selected."	Context is NOT maintained as users navigate the developer website.

## When is it a good idea to use zone pivots?

## NOTE

This section makes assumptions based on current functionality of the zone pivots feature. It is possible that in future releases, this guidance will change as gaps between tabbed conceptual and zone pivots close.

Start with the question:

*"Does the user have enough information to make a decision before reading the document?"*

If the answer is no, then zone pivots aren't the right choice, since zone pivots only appear once at the top of the

document (consider tabbed conceptual). If the answer is yes, then there are follow-up questions that you can ask yourself to help make an informed decision:

- Will the zone pivot configuration change that you're proposing be reusable? For example, if it's a collection of programming languages, will another service be able to use this same configuration or is this a one off, say to display features for a service? If the answer is no, these are not reusable, then zone pivots are probably **NOT** the right choice. Each new pivot group requires an update to a configuration group -- like the redirect file. If a new pivot group is created for one-off use cases, there is potential for this file to become unmanageable. Did I mention that no configuration is required for tabbed conceptual?
- Will the reader need to toggle between selections as they navigate through your document? If the answer is yes, then you don't want to use zone pivots, since the selection is made once within the document.
- Do you need to deep link into this document? If yes, once in the document, will the reader need to make a choice to toggle content? If the answer is yes, then zone pivots are **NOT** the right choice. You'll want to consider tabbed conceptual to help the reader orient themselves.

It may sound like we don't want you to use zone pivots. That's not the case. We're calling out use cases where you may want to consider another available option. So when is it a good time to use zone pivots? When the answer to the first question is **YES**. An example is quickstarts or tutorials with multiple language options. A user that selects "Python" or "C#" expects the guidance to be consistent throughout the document -- so, in this case, a zone pivot group is a good idea.

## Implementation principles

The goal of having principles is to ensure a consistent experience as zone pivots are made available for docs. These are a work in progress, and we will take into account any feedback that we receive that improves the experience for writers and our readers.

- Before creating a new zone pivot group, check to make sure that there isn't a group that you can reuse for your objective. Also, ask yourself:
  - Will using zone pivots help my readers/developers?
  - How will you track success?
  - What's the fallback plan if it doesn't work the way you expected?
- Create zone pivot groups that are reusable across multiple doc sets. This reduces the overhead associated with maintaining the `yml` configuration, and also puts the onus on writing teams to have discussions about expansion/updates not only for a single service, but as a collection of services. Think scalability!

## Limitations, considerations, and gotchas

There are some things that we need to keep in mind as we start to roll out zone pivots:

- In order to use zone pivots, a new `.yml` configuration file is required at the `/articles` level of your repository. This file is global and will be shared by all writers who publish from this repository.
- When a zone pivot group is referenced in an article, all the pivots in that group are rendered -- it's all or nothing. *For example: If you create a zone pivot group for programming languages that includes C#, Java, Python, Node.js, and Go, then reference it in your article, at the top of the page the reader will see an option to select C#, Java, Python, Node, or Go. This is true even if content is only provided for two of the five available pivots.*
- If multiple zone pivot groups share a similar intent, like programming languages, then they should use a consistent naming convention and order of precedence. This means that the group `title` and `prompt` should match, and that the pivot `id` and `title` for each entry should match. This ensures that when readers move through the docs, that the context of their previous choices follow them. *For example: if a reader selects Python in an article, if you follow the rules listed here, when the reader moves to another*

article where Python is available selection, it is automatically selected.

- If your docset is nested inside another docset (eg. `/visualstudio/mac/` is a separate docset to `/visualstudio/`) by default your pages will use the zone pivots defined in the root docset (ie. `/visualstudio/zone-pivot-groups.yml`). If you rely on this behavior, testing on `review.docs.microsoft.com` will be tricky as the root docset will not be included in your branch preview, and the pivots will fail to render until merged to *master*.

You can override this behavior with a `docfx.json` `globalMetadata` entry `zone_pivot_group_filename: mac/zone-pivot-groups.json` to cause your pages to use the pivots defined in the nested docset. The path is given relative to the root docset (eg. `mac/` in this case), and file extension is `.json` rather than `.yml` since that's how the pivot information is rendered live.

## Configure your repository

Before you continue, it's always a good idea to make sure that this file doesn't already exist. Do a quick search in your repo for `zone-pivot-groups.yml`. If this file exists, skip to [Create a zone pivot group](#). If this file doesn't exist, use these instructions to create a configuration and enable zone pivots for your repository.

### Update `.openpublishing.publish.config.json`

If you're the first person to try and use zone pivots in your repository, the first thing you'll need to do is update `.openpublishing.publish.config.json`.

1. Open `.openpublishing.publish.config.json`. It's located in the root of your repository.
2. Locate the `type_mapping` object, and add this line:

```
"ZonePivotGroups": "Toc",
```

#### TIP

Here's a complete example from the [azure-docs-pr](#) repository.

3. Don't forget to save.

### Create a zone pivot definition file

By default, the Docs build system expects a file called `zone-pivot-groups.yml` at the root of the docset. For example, in `azure-docs-pr` the root is the `articles` folder.

1. Create a file named `zone-pivot-groups.yml` as a direct child of your docset root folder, such as `articles` in `azure-docs-pr` or `docs` in `sql-docs-pr`. Note that the root folder is different for each repo depending on configuration, so contact your repo admin or Docs Support for help if needed.
2. This file will contain all of the pivot groups for your docset. Here's a sample that you can use as a starting point.

```

# YamlMime:ZonePivotGroups
groups:
- id: my-first-zone-pivot-group
  title: This is the title for your zone pivot group
  prompt: This text is rendered for the user above their selection
  pivots:
  - id: id-for-first-zone-pivot-selection
    title: First tab
  - id: id-for-second-zone-pivot-selection
    title: Second tab

```

3. Each group that you create can have multiple zone pivot groups, but each group has a minimum bound of two pivots and a max of five pivots.

4. Save.

Alternatively, you can create a definition file with a different name and/or at a different location in your repo. This can be helpful for managing zone pivots in large repos. If you use a definition file other than the default, you must reference it by metadata in every article you want to call it from - so this approach is only recommended for large repos with too many pivot groups to easily manage in one definition file. To create and use a custom zone pivot definition file:

1. Create a YAML file with a meaningful name somewhere under the docset root. For example, if you want a definition file specifically for Data Explorer articles in the azure-docs-pr repo, you might add it as a direct child of the `articles/data-explorer` folder, and name it something like `data-explorer-zone-pivot-groups.yml`.
2. Create zone pivot groups as described in the next section.
3. Add the `zone_pivot_group_filename` metadata to every article where you want to use these pivot groups.
  - For example, you might add the following to the YAML header in individual Markdown files:

```
zone_pivot_group_filename: data-explorer/data-explorer-zone-pivot-groups.json
```

#### **IMPORTANT**

Although the zone pivot definition source file is .yml, this metadata points to the published output file, which is .json.

- Or, you can add the value for an entire folder via docfx.json as described in [How to apply metadata to a folder or path within a docset](#):

```

"fileMetadata": {
  "zone_pivot_group_filename":
  {
    "articles/data-explorer/**/*.md": "data-explorer/data-explorer-zone-pivot-groups.json"
  }
},

```

## Create a zone pivot group

Follow these instructions to create a new zone pivot group.

1. In your repository, open `articles/zone-pivot-groups.yml` in your favorite text editor.
2. Make a copy of an existing group, then add the appropriate values for your task.
  - Make sure that you use a unique name for the group `id`.

- If a similar intent exists, make sure that you use the same `title` and `prompt` for your group.
  - If a similar intent exists, make sure that you use the same pivot `id`s and `title`s.
3. Don't forget to save!

Here's a sample configuration:

```
# YamlMime:ZonePivotGroups
groups:
- id: programming-languages-set-one
  title: Programming languages
  prompt: Choose a programming language
  pivots:
    - id: programming-language-csharp
      title: C#
    - id: programming-language-java
      title: Java
    - id: programming-language-python
      title: Python
    - id: programming-language-nodejs
      title: Node.js
    - id: programming-language-go
      title: Go
# Notice how the ids, titles, and prompts are
# consistent and in order.
- id: programming-languages-set-two
  title: Programming languages
  prompt: Choose a programming language
  pivots:
    - id: programming-language-java
      title: Java
    - id: programming-language-go
      title: Go
```

## Enable zone pivots in an article

After you've updated `zone-pivot-groups.yml`, the next step is to add the appropriate metadata tag to your article. Keep in mind that once you add this to your article, regardless of whether or not you actually add content to the zone pivots, the selector `will` render on the page.

1. Open an article.
2. Add this metadata tag, replacing the content in brackets with the group `id` from `zone-pivot-groups.yml`. Notice the metadata field name contains underscores, not dashes:

```
zone_pivot_groups: <zone-pivot-group-id>
```

3. Save.

## Add content to your zone pivots

For each pivot in the selector, you'll need to add a code block to your document where you want it to appear.

1. All pivots start with: `::: zone pivot=<pivot-id>"`
2. All pivots end with: `::: zone-end`
3. Any content that you want to render when that pivot is clicked should reside within those tags.
4. If you have a document where you need multiple (or specific sections) to pivot with each click/selection (for example: selecting C#), you can reuse the pivot `id`.

The syntax is illustrated in this sample:

```
::: zone pivot=<pivot-id>  
[!INCLUDE [Name-of-article-or-content](includes/<your-include>.md)]  
::: zone-end
```

**TIP**

This samples uses an include file for simplicity. However, you can use inline markdown if you prefer.

## Combining pivots

If needed, you can render content based on a selection of multiple pivots. This is done by including the specific pivots in a comma-separated list. For example, the following content would be rendered if either the Linux pivot or the MacOS pivot were selected:

```
::: zone pivot="linux,macos"  
Content specific to Linux and Mac.  
::: zone-end
```

## Sample implementation

Zone pivots have been implemented for Translator Text. For this service, we decided to organize each guide by programming language. Each language specific guide is in an include and referenced in the document. This approach was used because the sections and explanations are not uniform for all languages.

- [Link to sample article](#)

# Write about moving service resources to a different Azure region

4/16/2021 • 4 minutes to read

This article provides guidance for writing a standard, per-service article about moving service resources to a different Azure region.

Our customers need to move service resources between Azure regions for all kinds of business reasons. They might move to a newly available region, deploy features or services available only in a specific region, move due to internal policy or compliance requirements, or to solve capacity issues.

The need to move resources impacts most Azure services. To respond to this need, we aim to have:

- A per-service article that describes how to move service resources to another region.
- Consistency across these per-service articles.

## TOC

The article should be placed under the How-to guides section.

The exact location is service-dependent. We suggest a dedicated section under How-to guides. The section should be named **Move**, or **Move between regions**.

## Metadata

Make sure the following metadata is included in the article.

ATTRIBUTE	VALUE
ms.topic	how-to
ms.custom	subject-moving-resources

### Title attribute

Include the words "move", "region", and the service name in the title attribute.

### Description attribute

Include the words "move", and "region" in the description.

## Customer intent statement

Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

**Example:** "As an Azure service administrator, I want to move my service resources to another Azure region".

This statement provides a record of the intent of this article for future contributors. For more information, see

the [Metadata overview](#).

## H1 (Headline)

The article heading should be clear. Use the words "move" "region" in the H1.

Under the H1, write a short intro that sets context, to help customers understand why they might move resources to another region.

- Answer the fundamental question, "why would I want to do this?".
- The exact detail and scope of the intro will depend on your service.

### **Sample paragraph:**

"This article describes how to move service-name resources to a different Azure region. You might move your resources to another region for a number of reasons. For example, to take advantage of a new Azure region, to deploy features or services available in specific regions only, to meet internal policy and governance requirements, or in response to capacity planning requirements."

## Screen shots

If you think that screenshots will offer value in recognizing symptoms or resolving problems, include them. Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article. For more information, see [Screenshots: How to create, format, and embed in documentation](#).

## H2s and H3s

Don't number H2 and H3 headings. H2s and H3s are procedure-based and customers understand that the sections follow in sequence.

Follow the H2 headings with one or two transitional sentences explaining why the steps need to be performed, or how that section contributes to the whole.

## Prerequisites (H2)

This section is required. Make **Prerequisites** your first H2. Prerequisites include anything that must be in place before starting to move resources to another region.

Entries in this section might be requirements or limitations, checks that should be made, or steps that must be taken before the move. If there are no prerequisites, state that in this section.

## Prepare (H2) (required)

This section is required. Procedural instructions that describe how to prepare service resources before moving them.

The actual steps will depend on the solution your service uses to shift resources across regions.

## Move (H2)

This section is required. Procedural instructions that describe how to move service resources to another region.

For example, you might describe how to export a resource group from the source region, and then import it into the target region.

## Verify (H2)

This section is required. Procedural instructions to check that resources have been moved, and appear as expected in the target region.

## Commit (H2)

This section is optional. Add it if your service supports a commit action to complete the move after verification.

## Discard target resources (H2)

This section is optional, but customers might need these instructions if the move to the target region was a test or proof-of-concept, or if they no longer need the resources after they've been moved.

## Clean up source resources (H2)

This section is required. Procedural instructions for cleaning up and deleting source resources that now exist in the target region.

Steps include instructions to delete the source resources that were moved, and to delete any additional resources that were created specifically for the move.

## Next steps (H2)

This section is optional. If there are actions that should be taken now that resources are running in the new target region, provide links to information about those actions.

## Existing article examples

Examples of services that already have a region-move article in place. Note that some of these examples were written a while ago and might not exactly match the outline described in this article.

- [Move NSGs to another region](#)
- [Move public IP addresses to another region](#)
- [Move a storage account to another region](#)

## Questions

If you have questions or problems related to this guidance, or wish to request changes, please contact raynew.

# Provide more information about Azure load-balancing options

4/16/2021 • 2 minutes to read

This article provides guidance for articles that mention load balancing in Azure.

Our customers tell us they find it difficult to determine what load-balancing options are available in Azure. They need to know when and how to use the load-balancing options that are available.

Load balancing is a common customer requirement. We need to make it clear what Azure offers and how to choose the best service for their needs. To help customers understand Azure load-balancing options, link them to the load balancing overview article in all your articles that mention load balancing.

## Load balancing link

Include a link to the following article in all your articles that mention load balancing:

<https://docs.microsoft.com/azure/architecture/guide/technology-choices/load-balancing-overview>

## Existing article examples

The following Networking article has a Note with a link to the load-balancing options overview article:

<https://docs.microsoft.com/azure/traffic-manager/traffic-manager-overview>

## Questions

If you have questions or problems related to this guidance, or wish to request changes, contact victorh.

# Write a how-to article about planning for and managing costs for an Azure service

4/16/2021 • 5 minutes to read

This article is intended to help you write a how to article about how to plan for and manage Azure costs. There's also a markdown template that can go directly into your article about [planning to manage costs for an Azure service](#). The template has the `ms.custom: subject-cost-optimization` metadata entry. Be sure to keep that in your article. It's used to track horizontal work.

*Note for Azure service writer: You can review published examples at:*

- <https://docs.microsoft.com/azure/cosmos-db/plan-manage-costs>
- <https://docs.microsoft.com/azure/storage/common/storage-plan-manage-costs>
- <https://docs.microsoft.com/azure/machine-learning/concept-plan-manage-cost>

When you're ready to publish your article, create a table of contents entry in your How-to guides section where appropriate. Title it "Plan and manage costs".

The template has five main subject areas for you to work with:

## Estimate costs with the Pricing Calculator

Readers may have never used Azure or your service. If so, they might be unaware they can review estimated service costs before they use it. The section, as an H2, covers estimated costs shown in the Azure Pricing Calculator before they create any service resources. Add an Azure Pricing Calculator image that shows estimated costs for your Azure service. Here's an example:

The screenshot shows the Azure Pricing Calculator interface. At the top, there's a header with 'Your Estimate' and navigation links for 'Expand all', 'Collapse all', and 'Delete all'. Below the header, a section for 'Virtual Machines' is shown, with a note that the selected instance is 'A1: 1 cores, 1.75 GB RAM, 70 GB disk' at a cost of '\$66.96'. The main area is titled 'Virtual Machines' and includes filters for 'REGION: West US', 'OPERATING SYSTEM: Windows', 'TYPE: (OS Only)', 'TIER: Standard', and an unchecked 'ADD MANAGED DISKS' option. Below these filters, an 'INSTANCE:' dropdown shows 'A1: 1 Core(s), 1.75 GB RAM, 70 GB Disk, \$0.090/hour'. To the right of the filters, there are buttons for 'Clone' and 'Delete', and a 'More info' section with links to 'Pricing details', 'Product details', and 'Documentation'. In the center, there's a summary row with '1 Virtual machines' and '31 Days', followed by an equals sign and the total cost '\$66.96' which is highlighted with a red box. At the bottom, a 'Support' section shows 'Included' support at '\$0.00', and a summary row for 'Estimated monthly cost' at '\$66.96' also highlighted with a red box. This row includes a dropdown for 'US Dollar (\$)' and a 'Export' button.

## Understand the billing model for the new resource

The billing model for a service is everything that affects costs shown on a bill that relates to your service. The largest component is the use of resources for your service. However, there can be charges for other services that are related to your service.

### How users are charged for your service

Every paid Azure service has billable meters and there's a unit of measure for each meter. Tell the readers what the meters are for your service and what the unit of measure is for each one. At the end of the billing cycle, charges for each meter are summed. The customer's bill or invoice has a section for all costs for your service. There's a separate line item for each meter.

We need to tell users how they're charged for your service. We don't need great detail, at a minimum:

Tell users about the billable meters and units of measure for your service.

Billable meters are the individual components of your service that appear on the customer's bill and are also shown in cost analysis. They're sometimes shown in the Azure Pricing Calculator, too. We need to give a brief explanation about the meters used by your service. At a minimum, list the meters for your service and talk about the unit of measure for them.

A unit of measure varies greatly among Azure services. It could be:

- Time-based like seconds, minutes, hours, and so on
- Size based, KB, MB, GB, and so on
- Number of transactions

### Other costs

When creating or enabling a resource for an Azure Service, many customers might not be aware of additional infrastructure costs that they'll incur.

For example:

- When a user creates a resource for your service, does the new resource need *other* Azure service resources to function?
- Will the user need to manually create the other resources?
- Will the other resources get automatically created?

If yes to any of the questions above, provide users with information that lets them know about the other potential costs. You might need to sync with your product team to get more insight about the behavior of costs for your service.

Let's consider a virtual machines example. When a user creates a virtual machine, they might get charged for:

- Disk
- Networking
- Bandwidth
- Storage
- Potentially other Azure services

*Some* of the costs (not all) that might accrue for a VM are shown in the pricing calculator.

Pricing Calculator | Microsoft Azure

azure.microsoft.com/en-us/pricing/calculator/?&ef\_id=CjwKCAjwyd

Apps GitHub GH Issues-Billing Work Personal CostMgtBilling

Overview Solutions Products Documentation Pricing Training Marketplace Partners More

## Virtual Machines

REGION:

West US

OPERATING SYSTEM:

Windows

TYPE:

(OS Only)

INSTANCE:

D2 v3: 2 vCPU(s), 8 GB RAM, 50 GB Temporary storage, \$0.2

VIRTUAL MACHINES

1

### Savings Options

Save up to 72% on pay-as-you-go prices with 1-year or 3-year Reserved Virtual Machine Instances. I applications with steady-state usage and applications that require reserved capacity. [Learn more about pricing.](#)

#### Compute (D2 v3)

Pay as you go

1 year reserved (~32% discount)

3 year reserved (~57% discount)

\$85.41

Average per month  
(\$0.00 charged upfront)

#### OS (Windows)

License included

Azure Hybrid Benefit

\$67.16

Average per month  
(\$0.00 charged upfront)

▼ Managed Disks

▼ Storage transactions

We need to let users know about the potential costs for those items. So, if applicable, add the following items to the section:

- Information about the types of costs that customers should expect to see that result from creating a service resource.
- Call out any costs that might continue to accrue charges if the user deletes the resource that was originally created. In the example above, if the VM is deleted, the disk might not get deleted. If not deleted, the disk might continue to accrue charges for the user.
- Add a statement about any unseen costs that aren't shown in the pricing calculator or Azure portal creation experience that the customer might also incur. Continuing with the VM example above, the VM might accrue networking and IP address costs that aren't shown in the pricing calculator estimate.
- Let the user know if service charges can be fulfilled by Azure Prepayment credit, previously called EA monetary commitment.

## Review estimated costs in the Azure portal

As users create resources for your service in the Azure portal, most services show estimated prices. This section, as an H2, discusses costs shown in the Azure portal when resources are created. Add an image that shows estimated costs when an Azure resource is getting created. Here's an example:

The screenshot shows two windows side-by-side. On the left is the 'Create virtual machine' wizard, step 2: 'Size'. It lists three options: D1\_V2 Standard, D1 Standard, and A1 Standard. Each option includes a summary of resources (1 Core, 3.5 GB, 2 Data disks, 2x500 Max IOPS, 50 GB Local SSD, Load balancing), a price (98.95 USD/MONTH (ESTIMATED), 104.16 USD/MONTH (ESTIMATED), 66.96 USD/MONTH (ESTIMATED)), and a 'View all' link. A hand cursor is hovering over the 'View all' link for the A1 Standard row. On the right is the 'Choose a size' blade, which provides a detailed description of estimated prices.

**Create virtual machine**

1 Basics Done ✓

2 Size Choose virtual machine size >

3 Settings Configure optional features >

4 Summary Windows Server 2008 R2 SP1 >

**Choose a size**  
Browse the available sizes and their features

Prices presented are estimates in your local currency that include only Azure infrastructure costs and any discounts for the subscription and location. The prices don't include any applicable software costs. Recommended sizes are determined by the publisher of the selected image based on hardware and software requirements.

★ Recommended | [View all](#)

D1_V2 Standard	D1 Standard	A1 Standard
1 Core	1 Core	1 Core
3.5 GB	3.5 GB	1.75 GB
2 Data disks	2 Data disks	2 Data disks
2x500 Max IOPS	2x500 Max IOPS	2x500 Max IOPS
50 GB Local SSD	50 GB Local SSD	Load balancing
Load balancing	Load balancing	

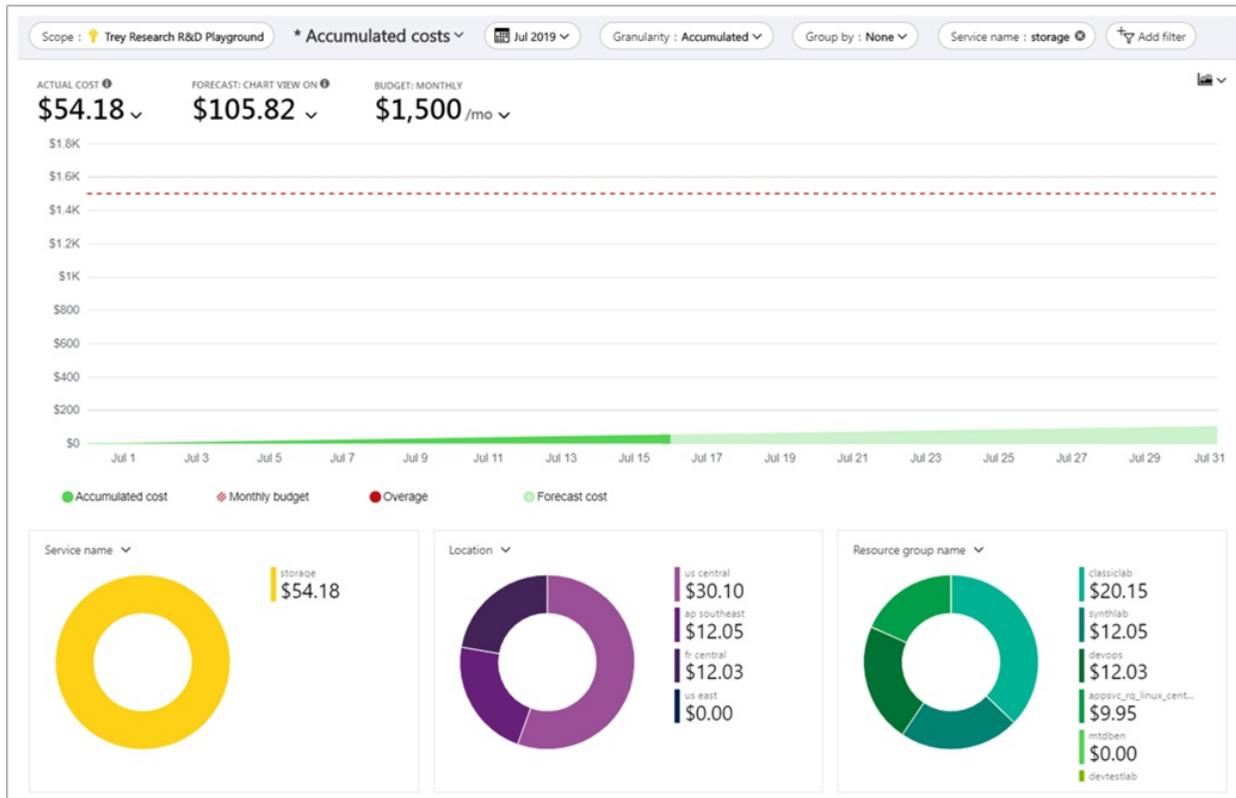
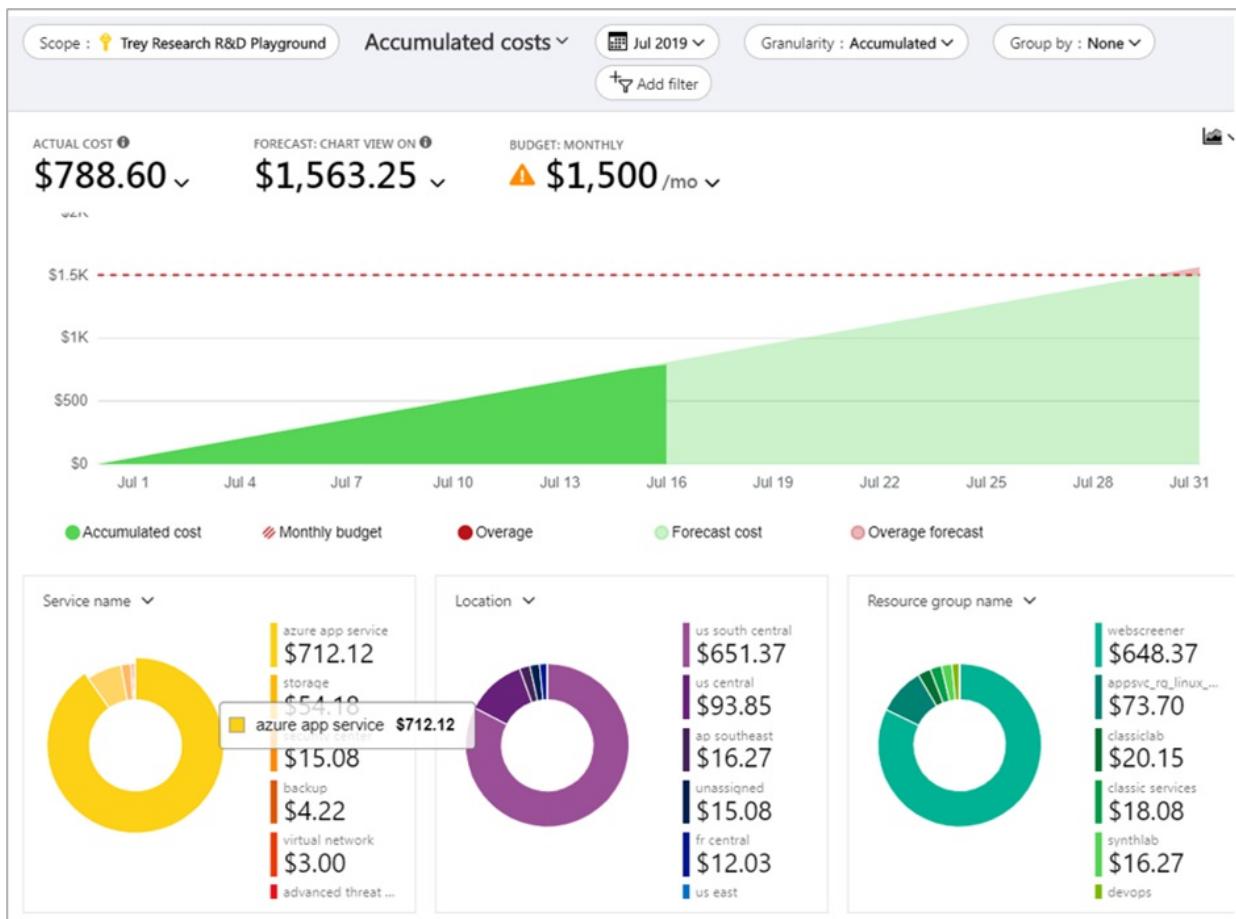
**D1\_V2 Standard** ★  
1 Core  
3.5 GB  
2 Data disks  
2x500 Max IOPS  
50 GB Local SSD  
Load balancing  
98.95 USD/MONTH (ESTIMATED)

**D1 Standard** ★  
1 Core  
3.5 GB  
2 Data disks  
2x500 Max IOPS  
50 GB Local SSD  
Load balancing  
104.16 USD/MONTH (ESTIMATED)

**A1 Standard** ★  
1 Core  
1.75 GB  
2 Data disks  
2x500 Max IOPS  
Load balancing  
66.96 USD/MONTH (ESTIMATED)

## Use Cost Management features

Cost Management has a number of features that help users view and manage costs. Budgets and associated alerts help users proactively manage costs. After users create resources, they can use cost analysis features to explore and manage costs. There are three H2s about Cost Management features in the template: **Monitor costs**, **Create budgets**, and **Export data**. Add two images: one that shows the total actual costs in cost analysis and one that shows costs for just your Azure service. Here are some examples:



Provide relative links to the following Cost Management articles that explain how to perform common tasks and use best practices and add the campaign ID found in the template:

- [Prevent unexpected costs](../cost-management-billing/understand/analyze-unexpected-charges.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn)
- [Monitor costs with Cost Analysis](../cost-management-billing/costs/quick-acm-cost-analysis.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn)
- [Guided learning for Azure Cost Management](/learn/parts/control-spending-manage-bills?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn)
- [How to optimize your cloud investment with Azure Cost Management](../cost-management-billing/costs/cost-mgt-best-practices.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn)

## Other ways to manage costs

You might already have published cost-saving content for your service. If feasible, move that content into this article or at least summarize key points and link to more details. When you have more comprehensive content, add links to your other published articles in this H2 section.

## Need help?

If you need help, contact BAnders.

# Write about Azure RBAC

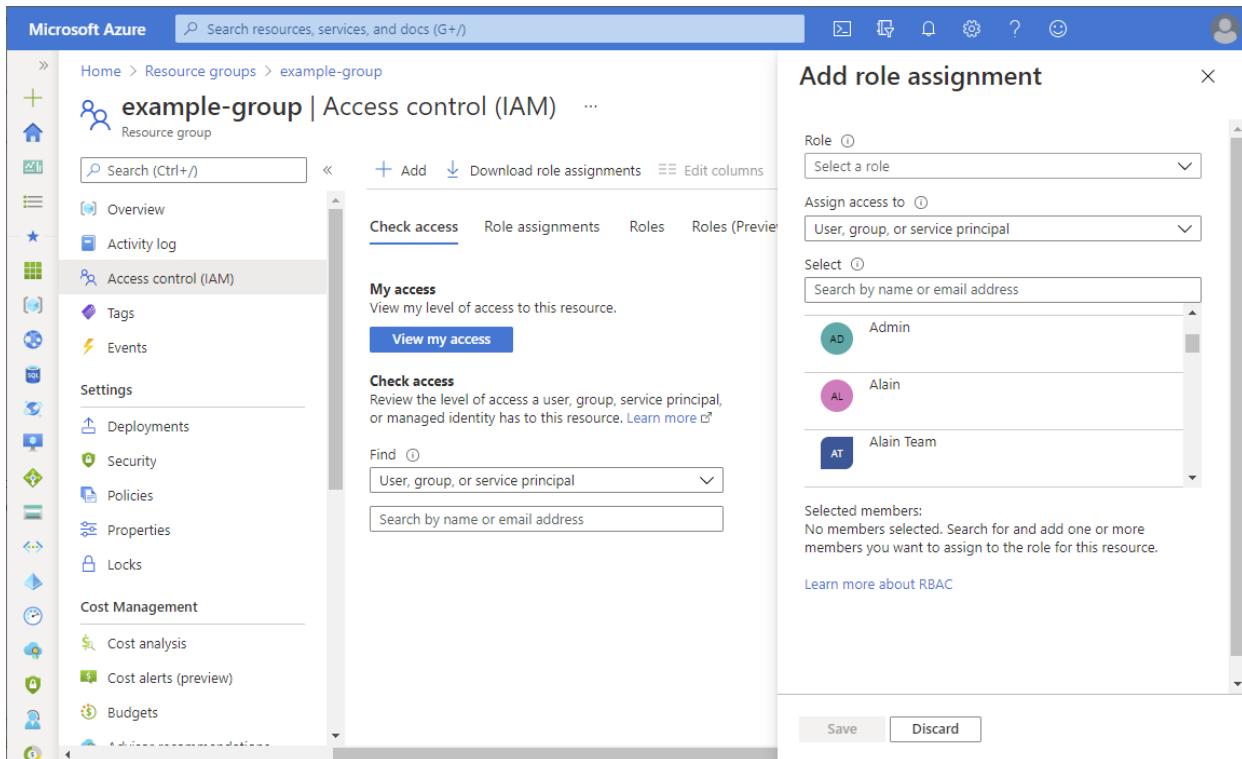
6/8/2021 • 5 minutes to read

Starting in April 2021, the experience to add role assignments in [Azure role-based access control \(Azure RBAC\)](#) will change. Instead of a pane, customers will use a full-screen page with tabs. If the articles you write have steps to assign Azure roles, you'll need to update them.

This article helps you write content to assign roles for Azure RBAC.

## What is changing?

The current role assignment experience uses a pane:



The new role assignment experience uses a full-screen page with tabs:

A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Owner	Grants full access to manage all resources, including the ability to a...	BuiltInRole	General	<a href="#">View</a>
Contributor	Grants full access to manage all resources, but does not allow you ...	BuiltInRole	General	<a href="#">View</a>
Reader	View all resources, but does not allow you to make any changes.	BuiltInRole	General	<a href="#">View</a>
AcrDelete	acr delete	BuiltInRole	Containers	<a href="#">View</a>
AcrImageSigner	acr image signer	BuiltInRole	Containers	<a href="#">View</a>
AcrPull	acr pull	BuiltInRole	Containers	<a href="#">View</a>
AcrPush	acr push	BuiltInRole	Containers	<a href="#">View</a>
AcrQuarantineReader	acr quarantine data reader	BuiltInRole	Containers	<a href="#">View</a>
AcrQuarantineWriter	acr quarantine data writer	BuiltInRole	Containers	<a href="#">View</a>

## What do I need to do?

Follow these steps to update your role assignment procedures.

1. Watch [meeting recording from April 22](#) that provides an overview.
2. Open the [Azure RBAC steps spreadsheet](#) to find which of your docs might be affected. See the **RBACSteps** tab.
3. Choose one of the three options in these guidelines to update your role assignment steps.
  - Option 1: Link to Azure RBAC docs
  - Option 2: Link to Azure RBAC docs with screenshot
  - Option 3: Add role assignment steps with screenshots
4. Update your docs.
5. Add `subject-rbac-steps` to your `ms.custom` metadata. We use this metadata and the spreadsheet to track completion.
6. Publish your updates.
7. Update the **Status** column in the [Azure RBAC steps spreadsheet](#) to indicate that you're done.

## When can I start?

You can start and publish anytime.

## When do I need to be finished?

The estimated general availability (GA) date is **June 30, 2021**. This date might change.

### Option 1: Link to Azure RBAC docs (Recommended)

To make it easier to maintain your docs, we recommend removing the role assignment steps and linking to the

Azure RBAC docs.

**Example:**

1. Assign the Virtual Machine Contributor role to the Alain user at the example-group resource group scope.

For detailed steps, see [Assign Azure roles using the Azure portal](#).

**Example Markdown:**

```
1. Assign the [ROLENAME] role to the [USER | GROUP | SERVICEPRINCIPAL | MANAGEDIDENTITY] at [MANAGEMENTGROUP | SUBSCRIPTION | RESOURCEGROUP | RESOURCE] scope.
```

```
For detailed steps, see [Assign Azure roles using the Azure portal](../role-based-access-control/role-assignments-portal.md).
```

## Option 2: Link to Azure RBAC docs with screenshot (Recommended)

To make it easier to maintain your docs, we recommend removing the role assignment steps and linking to the Azure RBAC docs.

**NOTE**

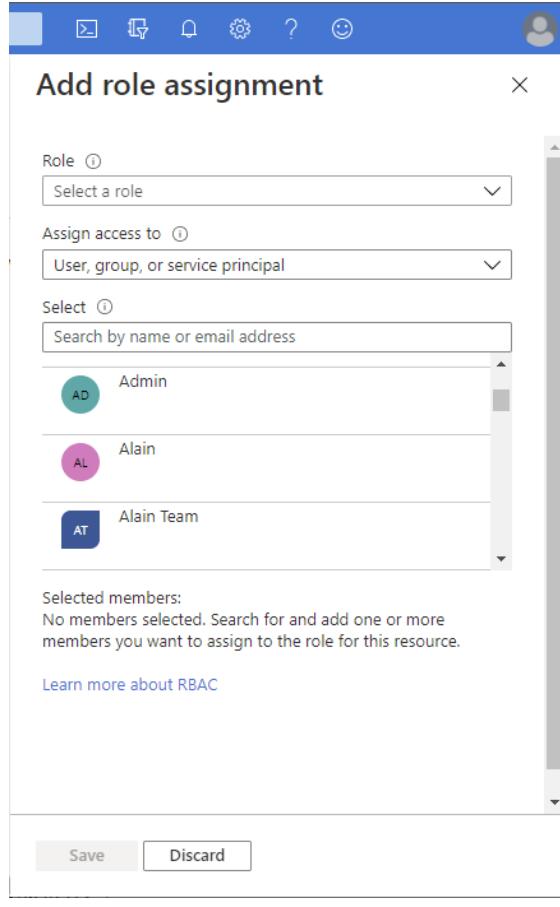
The screenshot (add-role-assignment-page.png) is an **include** that will change for GA and will be maintained.

**Example:**

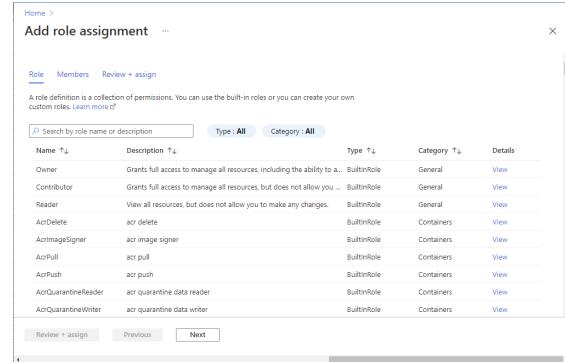
1. Select **Access control (IAM)**.
2. Select **Add > Add role assignment** to open the **Add role assignment** page.
3. Assign the following role. For detailed steps, see [Assign Azure roles using the Azure portal](#).

SETTING	VALUE
Role	Virtual Machine Contributor
Assign access to	User
Members	Alain

## BEFORE GA



## AFTER GA



### Example Markdown:

```
1. Select **Access control (IAM)**.

1. Select **Add** > **Add role assignment** to open the Add role assignment page.

1. Assign the following role. For detailed steps, see [Assign Azure roles using the Azure portal](../../role-based-access-control/role-assignments-portal.md).

| Setting | Value |
| --- | --- |
| Role | [ROLENAMESPACE] |
| Assign access to | [USER, GROUP, SERVICEPRINCIPAL, or MANAGEDIDENTITY] |
| Members | [MEMBERS] |

![Add role assignment page in Azure portal](../../../../includes/role-based-access-control/media/add-role-assignment-page.png)
```

## Option 3: Add role assignment steps with screenshots

If you assign a role in a quickstart or tutorial and you don't want to link the reader to another experience, use these steps. Notice that the screenshots are **includes** and will be maintained.

### NOTE

The screenshots (add-role-assignment-menu-generic.png and add-role-assignment-role-generic.png) are **include** files that will be maintained.

## Example:

The following steps describe how to assign the Virtual Machine Contributor role to the Alain user at the example-group resource group scope. For detailed steps, see [Assign Azure roles using the Azure portal](#).

1. Sign in to the [Azure portal](#) as [User Access Administrator](#) or [Owner](#).
2. Open the example-group resource group.
3. In the navigation menu, select **Access control (IAM)**.
4. Select **Add > Add role assignment (Preview)**.

A screenshot of the Azure portal's 'Access control (IAM)' page. The left sidebar shows 'Overview', 'Activity log', 'Access control (IAM)', 'Tags', and 'Events'. The 'Access control (IAM)' item is selected. The main area has a search bar and buttons for '+ Add', 'Download role assignments', and 'Edit columns'. A dropdown menu is open over the '+ Add' button, with 'Add role assignment' at the top, followed by 'Add role assignment (Preview)' (which is highlighted with a red box), 'Add co-administrator', and 'Add custom role'. At the bottom of the dropdown is a blue 'VIEW MY ACCESS' button.

5. On the **Role** tab, select the **Virtual Machine Contributor** role.

A screenshot of the 'Add role assignment' dialog. The top navigation bar includes 'Home', 'Add role assignment', and a close button. Below it are tabs for 'Role', 'Members', and 'Review + assign'. The 'Role' tab is selected. A message states: 'A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. [Learn more](#)'.

The main area is a table listing roles:

Name ↑↓	Description ↑↓	Type ↑↓	Category ↑↓	Details
Owner	Grants full access to manage all resources, including the ability to a...	BuiltinRole	General	<a href="#">View</a>
Contributor	Grants full access to manage all resources, but does not allow you to ...	BuiltinRole	General	<a href="#">View</a>
Reader	View all resources, but does not allow you to make any changes.	BuiltinRole	General	<a href="#">View</a>
AcrDelete	acr delete	BuiltinRole	Containers	<a href="#">View</a>
AcrImageSigner	acr image signer	BuiltinRole	Containers	<a href="#">View</a>
AcrPull	acr pull	BuiltinRole	Containers	<a href="#">View</a>
AcrPush	acr push	BuiltinRole	Containers	<a href="#">View</a>
AcrQuarantineReader	acr quarantine data reader	BuiltinRole	Containers	<a href="#">View</a>
AcrQuarantineWriter	acr quarantine data writer	BuiltinRole	Containers	<a href="#">View</a>

At the bottom are buttons for 'Review + assign', 'Previous', and 'Next'.

6. On the **Members** tab, select the Alain user.

7. On the **Review + assign** tab, select **Review + assign** to assign the role.

## Example Markdown:

The following steps describe how to assign the [ROLENAMESPACE] role to the [USER | GROUP | SERVICEPRINCIPAL | MANAGEDIDENTITY] at [MANAGEMENTGROUP | SUBSCRIPTION | RESOURCEGROUP | RESOURCE] scope. For detailed steps, see [Assign Azure roles using the Azure portal](../role-based-access-control/role-assignments-portal.md).

1. Sign in to the [Azure portal](https://portal.azure.com) as [User Access Administrator](../role-based-access-control/built-in-roles.md#user-access-administrator) or [Owner](../role-based-access-control/built-in-roles.md#owner).

1. Open the [MANAGEMENTGROUP | SUBSCRIPTION | RESOURCEGROUP | RESOURCE].

1. In the navigation menu, select \*\*Access control (IAM)\*\*.

1. Select \*\*Add\*\* > \*\*Add role assignment (Preview)\*\*.

![Access control (IAM) page with Add role assignment menu open.](.../includes/role-based-access-control/media/add-role-assignment-menu-generic.png)

1. On the \*\*Role\*\* tab, select the \*\*[ROLENAMESPACE]\*\* role.

![Add role assignment page with Role tab selected.](.../includes/role-based-access-control/media/add-role-assignment-role-generic.png)

1. On the \*\*Members\*\* tab, select the [USER | GROUP | SERVICEPRINCIPAL | MANAGEDIDENTITY].

1. On the \*\*Review + assign\*\* tab, select \*\*Review + assign\*\* to assign the role.

## Metadata

Include the following metadata when you update your add role assignment steps. We use this metadata and the spreadsheet to track completion.

ATTRIBUTE	VALUE
ms.custom	subject-rbac-steps

## Terminology

There are multiple RBAC systems, which can be confusing to customers. Always prefix RBAC with the appropriate modifier, such as Azure or Kubernetes. Avoid using RBAC on its own.

Starting in July 2020, we're updating Azure RBAC terminology across the Azure docs repo. Writers don't need to do anything, but you'll see updates in your docs. For a list of the terminology updates, see the [Azure RBAC Terminology spreadsheet](#).

When you describe Azure RBAC, use the following terminology:

RECOMMENDED TERM	AVOID
Azure role-based access control (Azure RBAC)	role-based access control (RBAC)
Azure RBAC	RBAC
Azure role assignment -or- role assignment	RBAC assignment

RECOMMENDED TERM	AVOID
Azure role -or- role	RBAC role
<b>Azure built-in role</b> -or- built-in role	built-in RBAC role
<b>Azure custom role</b> -or- custom role	custom RBAC role

For other RBAC systems, use the following terminology:

RECOMMENDED TERM	DESCRIPTION
Azure AD role	When referring to <a href="#">Azure Active Directory roles</a> .
resource local RBAC	A service-specific local RBAC that isn't Azure RBAC. A subset of resource local authorization that can include things like Key Vault access policies, Azure Files NTFS permissions, or Blob Storage POSIX ACLs.
Managed HSM local RBAC	When referring to local RBAC for <a href="#">Managed HSM</a> .
Kubernetes RBAC	When referring to <a href="#">Kubernetes RBAC</a> .
Service Fabric RBAC	When referring to <a href="#">Service Fabric RBAC</a> .
vSphere RBAC	When referring to <a href="#">vSphere RBAC</a> .

## Questions

If you have any questions or need help, contact Robert Lyon.

# Monitoring content for Azure services

3/31/2021 • 8 minutes to read

The purpose of the Azure monitoring horizontals project is to provide consistent information for customers to monitor the health and availability of Azure resources across all services. If you have questions on the content, please contact Rob Boucher ([robb@microsoft.com](mailto:robb@microsoft.com)) for more information.

## Services included

Most Azure services use [Azure Monitor](#) to collect and analyze their metrics and logs. The services that do not are being onboarded over time.

If your service uses Azure Monitor, then you're highly encouraged to create and maintain the content defined in this article for the reasons explained below.

If your service does not use Azure Monitor, we still recommend that you use the following Azure Monitor templates. We recommend that you create two articles as described below and then approximate the same flow, modifying the templates when headings and sections are not applicable. If your service will remain active for the foreseeable future, it's likely you will be onboarded to Azure Monitor. Regardless, you are currently still likely to have your own metrics, logs, and reference material. We won't enforce the use of the template and this guidance for services not using Azure Monitor, but doing this work now is likely to save you time in the future.

## Background

Azure Monitor provides a consistent set of tools and processes for collecting and analyzing monitoring data across all Azure services. Processes such as using metrics explorer to analyze metrics, configuring the collection of resource logs, or writing a log query are identical for all services.

Few customers use only a single Azure service. As such, they can leverage the knowledge and skills for monitoring a single service across every service that they use. If you know the shortcut keystrokes in Word, you can apply most of them to all of Office and the Windows OS. Azure Monitor is similar. We want customers to know when they're using a feature of Azure Monitor as opposed to a feature of a particular service so they understand that they can leverage that information to monitor other services or multiple services together.

To drive this requirement, we want to minimize duplication of content and refer to Azure Monitor for common processes. If a service documents a complete walk-through for creating a diagnostic setting to collect resources logs, for example, it's difficult for the customer to understand whether your content is just an example of a standard process or if it's a process specific to the particular service.

We do understand though that different services have different requirements, so the template does allow the flexibility for each service to provide examples and potentially screenshots specific to their service. Ensure though that any such content includes the link to the appropriate generalized Azure Monitor content and clearly identify it as an example of a standard process.

## Articles

The following table lists the articles that provide monitoring information for each Azure service. The details of each are described below. At a minimum, it would be great just to have these two articles exist for each service even if the format of the articles varies.

### Articles you write or update

ARTICLE TITLE	FILENAME NAME	LOCATION	COMPLETED BY	ACTION
<i>Monitoring &lt;Service Name&gt;</i>	monitor-<service-name>.md	Your service folder and TOC	You, the content writer for that service	The main article people should find in search when they want to monitor your service. For more information, see <a href="#">below</a> .
<i>Monitoring &lt;Service Name&gt; data reference</i>	monitor-<service-name>-reference.md	Your service folder and TOC	You, the content writer for that service	An article for additional reference information that is too long for your main monitoring article. See <a href="#">below</a> .
Common and service-specific schema for Azure Resource Logs	<a href="#">resource-logs-schema.md</a>	Azure Monitor	Azure monitor team. <b>Links updated by you.</b>	<b>UPDATE THIS ARTICLE.</b> The Azure monitor team maintains the top-level schema. Add your service to the table with a pointer to docs describing your specific log schema in the <a href="#">service specific schema section</a> . If such an article(s) do not already exist, put the schema information into <i>Monitor &lt;servicename&gt; data Reference</i> article and refer to that section.

## Articles you refer to and maybe update

The following articles support the articles above.

TITLE	NAME	TOC LOCATION	COMPLETED BY	DESCRIPTION	ACTION
Monitor Azure resources with Azure Monitor	<a href="#">monitor-azure-resource.md</a>	Azure Monitor	Azure Monitor content team	An article containing common information for services onboarded to Azure Monitor. Included in the <i>Monitoring &lt;Service Name&gt;</i> template text already. See <a href="#">more information below</a> .	No action

Title	Name	TOC Location	Completed By	Description	Action
Resource providers / types relevant to your service	<a href="#">azure-services-resource-providers.md</a>	Resource Manager	Resource Manager team.	A mapping of service to namespace (resource provider and type)	<b>You check.</b> Verify the article is updated for your service namespaces. If it isn't, correct and notify the author via a PR.
Azure Monitor supported metrics	<a href="#">metrics-supported.md</a>	Azure Monitor	Azure Monitor content team	Azure monitor team maintains this article. It's automated and pulled from backend systems.	<b>You check.</b> Verify that this article contains your metrics and refer to them in the <a href="#">reference article you write</a> via bookmarks. If there are incorrect descriptions, contact your PM team and have them update them in the Azure Monitor "shoebox" source. A developer must do the work in most cases. Contact Rob Boucher ( <a href="mailto:robb@microsoft.com">robb@microsoft.com</a> ) if you need more information.
Supported categories for Azure Resource Logs	<a href="#">resource-logs-categories.md</a>	Azure Monitor	Azure Monitor content team	Azure monitor team maintains this article. It's automated and pulled from backend systems.	<b>You check.</b> Verify that this article contains your log categories and refer to them in the <a href="#">reference article you write</a> via bookmarks.
Resource provider operations	<a href="#">Resource Provider Operations</a>	Resource Manager	RBAC writer team	A list of operations that could appear in the Activity log sorted by "service group"	No action.
Azure Activity Log event schema	<a href="#">activity-log-schema.md</a>	Azure Monitor	Azure Monitor content team	A standardized reference to entries added to the activity log.	No action.

Title	Name	TOC Location	Completed By	Description	Action
Azure Monitor Logs / Log Analytics Table Reference	<a href="#">Many articles</a>	Azure Monitor Reference	Azure Monitor content team	An automated reference of all the tables and descriptions available in Log Analytics. Link to the tables relevant to your service in your <i>Monitor &lt;servicename&gt;</i> or <i>Monitor &lt;servicename&gt; data Reference</i> article.	No action.

## Monitoring service templates

See [Azure monitoring templates](#).

### Monitoring [service] article

The **Service monitoring** article should be completed based on the template **monitor-service-template.md**. It follows a similar format as the [Monitoring Azure resources](#) article and includes a note recommending the user first review that article if they aren't familiar with Azure Monitor.

Your service monitoring article is intended to provide the following value:

- **Awareness of Azure Monitor to the customer** - Most services have a **Monitoring** UI element and screen in their menu in the Azure portal. We assume that many users will look to the content for that service for an explanation on that topic. The [Monitoring Azure resources with Azure Monitor](#).
- **Any uniqueness for the service** - Some services may have functionality beyond standard Azure Monitor. You can provide that information in this article and also clearly distinguish between information that only applies to your service from information that applies to all services.
- **Examples for the service** - The core Azure Monitor content provides concepts and processes that apply to all Azure services. You can choose to add specific examples for your service to your article. Examples include screenshots of the metrics explorer for metrics of the service or a command-line examples for configuring diagnostic settings.
- **Prescriptive guidance** - The core Azure Monitor content can provide general concepts of monitoring and specific processes to configure and analyze monitoring data. However, every service has different requirements in terms of what conditions to monitor. Include your specifics in your article. Examples include log queries to return valuable information, particular metrics to most accurately assess performance of the resource, and the definition of alert rules to proactively identify common issues.
- **Pointers to Reference information** - *Point* to reference content in your **monitoring service data reference** article. Don't relist it here if you can avoid it.

### Monitoring [service] data reference article

This article is for long lists of reference information. If you repeat reference information found in the above automated sources, you are responsible for keeping it updated and in sync with the articles that are maintained by the Azure Monitor and Azure Resource Manager writing teams. Those articles are listed earlier in the above table.

#### **NOTE**

Ideally, don't repeat information that is already in existing reference articles *UNLESS* you are adding additional information not in those articles. Added information includes further explanation. Many of the articles above are automated and new information appears after each run. If you find errors in the articles It's best that the articles aren't out-of-sync with each other.

## Monitoring Azure resources article

The [Monitoring Azure resources with Azure Monitor](#) article is intended to support the service monitoring articles. The intent of this article is to provide a starting point to a user who wants to monitor any Azure service but who has minimal initial knowledge of Azure Monitor. It quickly cuts through the Azure Monitor features that are relevant to monitoring an Azure service and links to articles with detailed concepts and processes.

## Getting help

If you have questions, feel free to email [Azure Monitoring Doc Writers](#) and explain your issue. You can get help with the templates, suggest modifications to the templates, get help collecting the proper information for your service if your PMs don't provide it, or even get help with understanding Azure monitor if you are confused about it after reading through the [Azure Monitor documentation introductory articles](#).

## Examples

See the following examples for public monitoring articles.

### **Azure Cosmos DB**

- [Monitoring Azure Cosmos DB](#)
- [Azure Cosmos DB monitoring data reference](#)

### **Machine Learning**

- [Monitoring Azure Machine Learning](#)
- [Azure Machine Learning Monitoring Data Reference](#)

### **Azure Storage**

- [Monitoring Azure Blob Storage](#)
- [Azure Blob Storage monitoring data reference](#)

# How to refer to public container images in content

5/5/2021 • 8 minutes to read

This guidance is for articles or samples that require a public container image. "Public" means an image hosted in a public registry such as Docker Hub and that gets pulled or referenced directly as part of an example.

If your example refers to *any* container image hosted in [Docker Hub](#), decide how to update it because of impacts from Docker's updated [Terms of Service](#). If your example uses container images from other public sources, updates are optional but recommended in several cases.

## Summary

Find your scenario in the following table and see the linked guidance for details.

IMAGE USED	EXAMPLE - <a href="#">MORE INFO</a>	DO THIS
First-party Microsoft image from Docker Hub	<code>microsoft/dotnet:2.1-runtime</code>	Replace with an equivalent image in Microsoft Container Registry - <a href="#">more info</a>
Third-party image from Docker Hub	<code>node:10.15.0</code>	Add guidance to maintain the image in a private Azure container registry - <a href="#">more info</a>
Docker Hub image metadata only (name, tags, etc.)	<code>https://hub.docker.com/_/mysql</code>	No change if the image isn't consumed
Image from Microsoft Container Registry	<code>mcr.microsoft.com/dotnet/core/sdk:2.1-stretch</code>	No change
Image from another public source other than Docker Hub	<code>public.ecr.aws/bitnami/tensorflow-resnet:latest</code>	Optionally add guidance to maintain the image in a private Azure container registry - <a href="#">more info</a>
Private image pushed/pulled to a private Docker Hub repository	<code>mydockerid/myspecialimage:tag</code>	No change if a recommended scenario for your product/service.  Optionally modify example to manage image in an Azure container registry.

## (TL;DR) Why this guidance is needed

Increasingly, container images are becoming a way for developers to develop, distribute, and deploy software and services. You might need to use a container image in one of your articles or scenarios, even if your product or service isn't strictly container-related.

Follow this guidance when your content includes an example that [pulls](#), [runs](#), or [refers to](#) a *public* container image or a related format such as a Helm chart that references public images. Public registries include:

- [Docker Hub](#)
- Public registries hosted by [AWS](#) or [GCP](#)
- Microsoft Container Registry - MCR is Microsoft's container registry for public pull access to (mostly) first-party images.

## What's the problem with Docker Hub?

Docker Hub has been a de facto standard location for hosting public container images. In November 2020, Docker changed its [Terms of Service](#) in a way that could throttle customers or services making anonymous pull requests to Docker Hub or requests using the Free account. This change could affect Docs customers who follow examples that pull public container images from Docker Hub, especially if those examples don't include steps to authenticate to Docker Hub.

[Industry guidance](#) is also moving to encourage image consumers to maintain private copies of any public image content they use, regardless of the source.

## Image reference examples

Content can refer to container images in different contexts. A container image name is of the form

<registryServer>/<repositoryName>:<tag>

- Fully qualified **registry server name**, such as `somepublicregistry.com`

If not provided, the default registry is Docker Hub (sometimes named explicitly as `docker.io`)

- **Repository name** in the registry, with optional namespace, such as `/path/to/repository`

- Optional **tag** for image, such as `:version2`

If not provided, the default tag is `:latest`

## Examples

CONTEXT	EXAMPLE
<code>docker pull</code> or <code>docker run</code> command	<code>docker run mcr.microsoft.com/azure-sql-edge</code>
<code>FROM</code> statement in a Dockerfile	<code>FROM mcr.microsoft.com/azure-functions/python:2.0-python3.7-appservice</code>
<code>image:</code> statement in Docker Compose or Kubernetes deployment file	<code>image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine</code>

## Do you use a Microsoft image from Docker Hub?

If your example directly refers to an image in the Microsoft organization in Docker Hub, you'll need to change it. These references appear like: `microsoft/<someRepo>:<someTag>`.

Example: `FROM microsoft/dotnet/core:latest`

**Mandatory guidance: Update the example to use an MCR image. Microsoft Container Registry is the only registry you should reference in content to pull a first-party Microsoft image.**

To find and substitute an appropriate `image:tag` reference in MCR (if available), see [How to search images in MCR](#).

### IMPORTANT

There isn't a one-to-one correspondence between names and tags of first-party images in Docker Hub and MCR, and some Microsoft images haven't been migrated from Docker Hub to MCR. Check image names carefully.

## Other guidelines

- Retest content whenever you substitute an MCR image or use a different tag than the one originally used

in the article or sample. Don't assume a different tag will work identically.

- If a Microsoft-owned image isn't in MCR, work with your engineering team to [onboard it](#).
- Some Microsoft teams develop images published to Docker Hub repos outside the Microsoft organization. Again, if an equivalent image isn't maintained in MCR, work with your engineering team to [onboard it](#).
- If a Microsoft image can't be onboarded to MCR or a team can't commit to maintain it, modify the example to use a different image from MCR, or deprecate the content.

## Do you use a third-party image from Docker Hub?

Third-party here means a public image not owned by Microsoft, for example an OSS image such as `mysql`.

Third-party images often are used as base images for Docker image builds, providing a base operating system or application framework.

### Update options

1. Is there an equivalent image already in MCR? This won't be a common case, but if so, substitute it. See [How to search images in MCR](#).
2. Can you change the example to use a different image from MCR? You might need input from your engineering partners to make this decision.
3. Does image meet criteria for [onboarding to MCR](#)?
4. If MCR isn't the solution, then see [How to add guidance for Azure Container Registry](#)

### Other guidelines

- Retest content whenever you substitute an MCR image or use a different tag than the one originally used in the article or sample. Don't assume a different image or tag will work identically.

## How to search images in MCR

MCR doesn't have a search GUI. However, you can use API commands to get the information you need:

OPERATION	COMMAND
List all repositories in the MCR Catalog	<code>https://mcr.microsoft.com/v2/_catalog</code>
List tags for a specific repository	<code>https://mcr.microsoft.com/v2/&lt;repoName&gt;/tags/list</code> Example: <code>https://mcr.microsoft.com/v2/dotnet/framework/aspnet/tags/list</code>

### TIP

It's not intuitive, but most MCR images have Readme content "syndicated" to Docker Hub to help customers discover the images. But the actual pull should always reference [mcr.microsoft.com](https://mcr.microsoft.com).

## How to onboard image to MCR

Onboarding requires commitments from the image owner for both initial setup and ongoing maintenance. Onboarding *may* be appropriate for certain third-party images that are required for product scenarios. However, this is the exception, not the rule. If a Microsoft or other candidate container image isn't already hosted in MCR,

work with your engineering team to onboard it. Alternatively, consider adding guidance to use [Azure Container Registry](#) to avoid Docker Hub rate-limiting.

If the image was developed solely for documentation, it might be appropriate to host in an MCR repo maintained by C+L. Reach out to Mangesh Sangapu for onboarding options. See detailed [MCR onboarding steps](#).

## How to add guidance for Azure Container Registry

If an article or sample must use a public non-Microsoft container image from Docker Hub, add guidance to host a copy in an [Azure container registry](#). Adding this guidance helps customers maintain a specific image in an environment they control and maintain.

Depending on the scenario, include some or all of the following guidance:

- Create a registry
- Import the image from Docker Hub
- Authenticate to the registry
- Update image reference to consume the image from that registry

It's straightforward to incorporate this guidance and adjust it for your scenario with a few commands using the Azure CLI. Details are in the following sections.

### NOTE

Adding this guidance is optional when examples use container images from public registries *other than* Docker Hub (such as Quay or Amazon ECR Public Gallery), because these registries don't have the same limitations as Docker Hub. However, consider adding the guidance for scenarios involving frequent image pulls.

### Create an Azure container registry

This step can be a prerequisite, a separate H2 heading, or an inline procedure. Many container-related scenarios already require a private container registry and include a prerequisite to create one.

- [Doc reference](#)
- [Include file with H2 heading](#)

### Create a resource group

```
az group create --name myResourceGroup --location <location>
```

### Create a container registry

```
az acr create --resource-group myResourceGroup \
--name myregistry --sku Basic
```

### Import image to the registry

This step can be a separate H2 heading or an inline procedure. Import is a one-time operation.

Replace the source `repo:tag` for the image from Docker Hub used in the following example. The image and tag names in the target registry are up to the customer. It's recommended for customer to provide Docker Hub credentials in the import command.

- [Doc reference](#)
- [Include file with H2 heading](#)

## Import image from Docker Hub (authenticate to Docker Hub)

```
az acr import \
--name myregistry \
--source docker.io/library/repo:tag \
--image myprivateimage:mytag \
--username <Docker Hub username> \
--password <Docker Hub token or password>
```

### Authenticate to registry for pull access

Authentication to the Azure container registry is required for all operations including pulling an image. You might need to add an H2 heading or inline procedure to login or to create credentials to allow pull access. Among the [ACR authentication options](#), ones most likely to add in this context are below.

#### Login to ACR

- [Doc reference](#)
- [Include file with H2 heading](#)

Direct login to the registry using the `az acr login` is the cleanest option if it works for your scenario, since the registry handles credentials for subsequent Docker pull operations.

```
az acr login --name myregistry
```

#### Repository-scoped token

A repository-scoped token is another option, because it can be granted minimal permissions scoped to a specific repository. Tokens can also be assigned to environment variables for use in later procedures and then used in a `docker login` command.

##### Doc reference

**Example:** Token with pull access (`content/read` action) to a specific repository, such as `hello-world`.

```
TOKEN_NAME=mytoken # token name can be any you choose
TOKEN_PW=$(az acr token create \
--name $TOKEN_NAME \
--registry myregistry \
--repository hello-world content/read \
--query credentials.passwords[0].value \
--output tsv)
```

Then, add a `docker login` command for customer to login to the registry:

```
echo $TOKEN_PW | docker login --username $TOKEN_NAME --password-stdin myregistry.azurecr.io
```

#### Access from AKS cluster

If you are using AKS with an [attached ACR](#), the cluster's managed identity is configured by default with pull access to the registry.

#### Replace reference to image

The last step is to update image references in content to refer to the image in the private registry instead of the image in Docker Hub.

Provide the fully qualified image name referring to the private Azure container registry. Example:

```
myregistry.azurecr.io/path/to/myimage:mytag .
```

**IMPORTANT**

The registry name and image repository name may contain only lowercase letters, numerals, and some special characters. However, the tag is case sensitive and may contain uppercase letters.

## Existing article examples

- [Push your first image to Azure container registry](#) - substitutes MCR image

## Next steps

If you have questions or problems related to this guidance, contact danlep or George Wallace's team.

# When to create a Docs troubleshooting article

6/16/2021 • 3 minutes to read

A troubleshooting article should help resolve a specific error or problem that customers commonly run into and would search on. Before you consider creating an article or creating a content request with Customer Support Services (CSS), ask yourself these questions:

- Would the customer run into the issue if we provided steps to avoid it in a conceptual or procedural article?
- Is it an issue that customers come across frequently as they're completing the steps in that article?

Are you not sure about the answers to those questions? Do you think the issue requires a separate troubleshooting article:

- Content developer/writers: Consult first with the CSS Supportability PM and the content project manager for your product or service area.
- Other roles like PM or Dev: Create a content request with CSS by using <http://contentidea>. For more information, see the [Content Idea user guide](#).

If you think it makes sense to add steps to avoid the issue into an existing article, [edit the article and create a pull request](#).

## Before you create an article

Before you create a troubleshooting article:

- Do a search on Google to see what articles come up for that issue. Is there already a solution available on a Microsoft property?
- Is there a related conceptual or procedural article? Would make sense to tell the customer how to avoid the issue in that article? Consider adding an H2 section at the end of the article for **Common Problems** or **Troubleshooting**.
- Is the issue a result of a content gap? Do we need to create a best practice article, checklist, or some other artifact that would prevent the customer from coming across the issue?
- Work with your CSS Supportability PM or technical advisor to confirm a need for a troubleshooting article. Target the article to deflect any of these top support issue types:
  - Highest number of cases.
  - Most costly for Customer Support Engineering to solve.
  - Strategically important to Microsoft to solve.
- Decide on the format of the article. To help you decide which format you need, see [Choose the right format for a troubleshooting article](#).

## Appropriate for Docs vs. not

It may be ok to include a troubleshooting article on Docs where all these things apply:

- You talked with CSS and agree that a KB article isn't appropriate for the issue.
- You'll write the article to focus on a customer intent and it will address a specific issue.
- Customers are commonly searching for help with the issue.
- Content can be public-facing.

- Issue applies to all regions/markets.

In Docs, don't include:

- Internal support content.
- Edge case issues.
- Market or region-specific issues.

## Format and location

Based on studies Content & Learning has done (and anecdotal information), customers prefer troubleshooting content that's targeted to resolve a specific issue. So think about customer intent for your titles and headers and use [SEO best practices](#).

Because troubleshooting articles are most often procedural, follow general style guidelines for procedures. You should add general troubleshooting articles under a **Troubleshoot** TOC node under the **How-to guides** TOC node. You should add problem resolution articles under a **Problem resolution** TOC node under the **Troubleshoot** node. If the content provides information for setting up tools or log collection, it would be more suitably located directly under the **How-to guides** node of the TOC.

## Measure effectiveness

Historically, many teams have found that troubleshooting articles don't do well, which is why we don't have many on Docs. It can be a bit tricky to figure out whether a troubleshooting article is helping customers.

Things to look at after you've published a troubleshooting article on Docs:

- From the [MDM tool](#):
  - Page views
  - CSAT
- From GitHub
  - Feedback / [Git Hub issues](#)
  - [Public PRs](#) for that the article
- From the CSS Supportability PM or content project manager for your area:
  - Support case deflection rate
  - Changes to support cases time to resolve
- Search results on [DuckDuckGo](#) or browser in-private browsing for the related keywords

Over time, if the metrics don't look good and aren't improving, you might consider a rewrite or retirement.

# Choose the right format for a troubleshooting article

6/4/2021 • 2 minutes to read

Use the resources in the following table to identify the appropriate format for the information you want to provide.

FORMAT	WHAT IS IT	WHEN TO USE IT
General troubleshooting	<b>Optional:</b> A procedural format that describes how to complete general troubleshooting steps for a service or feature.	Help customers resolve any problems or symptoms they're experiencing when they don't have a specific error message. Located under <b>How-to guides</b> -> <b>Troubleshoot</b> .
Problem resolution	<b>Optional:</b> A procedural format that describes how to resolve a specific problem.	Help customers solve a problem when they're given a specific error message. Located under <b>How-to guides</b> -> <b>Troubleshoot</b> -> <b>Problem resolution</b> .
Known issues	<b>Optional:</b> A list format that provides known issues for a feature or service.	Provide information about issues that are currently being worked on or planned to be fixed soon. Located under <b>How-to guides</b> -> <b>Troubleshoot</b> .
Error code lists	<b>Optional:</b> A list format of the error codes associated with the feature or service.	Inform about the errors that can occur and provide help in resolving them. Located under <b>Reference</b> .

# Write general troubleshooting articles

5/10/2021 • 3 minutes to read

General troubleshooting articles are written when a specific error message isn't known. The customer has encountered an issue that needs to be resolved without being clear about what is causing the issue.

You can consider categorizing the troubleshooting content for general troubleshooting into levels of expertise. For example:

- Level 100: Basic
- Level 200: Moderate
- Level 300: Moderate advanced
- Level 400: Advanced

Advanced levels require some experience with detailed troubleshooting methods. If using this structure, you should put each level of content in a separate topic for better comprehension by the intended audience. For example: [Resolve Windows 10 upgrade errors : Technical information for IT Pros](#).

Use this [general troubleshooting template](#) to get started. If you aren't sure whether you should be writing this type of article, see [Choose the right format for a troubleshooting article](#).

## TOC

If troubleshooting content is to be provided in a TOC, it should be located in a **Troubleshoot** node under the **How-to guides** node. General troubleshooting articles are located directly under the **Troubleshoot** node. Unless it is unavoidable, make sure your node titles don't wrap to a second line. For more information, see [TOC structure](#).

## Metadata

Make sure that the `ms.topic` attribute is set to `troubleshooting`. You should also include the word **troubleshooting** somewhere in the title and description attributes. Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This statement provides a record of the intent of the article for future contributors. For more information, see the [Metadata overview](#).

## H1 (Headline)

The heading of the general troubleshooting article should concisely describe the issue that is trying to be resolved. You should also include the word **troubleshoot** somewhere in the H1 of the article to assist in search engine optimization.

## Introduction

The article should begin with an explanation of the issue and what the customer should expect to see in the article. The information in the introduction should help the customer decide whether the information applies to the issue that they are encountering.

## Screen shots

If you think that screenshots will offer value in recognizing symptoms or resolving problems, include them. Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article. For more information, see [Screenshots: How to create, format, and embed in documentation](#).

## H2s and H3s

Don't number H2 and H3 headings. Follow the H2 headings with one or two transitional sentences explaining why the steps need to be performed or how that section contributes to the whole. When possible you should provide the following sections:

- **Prerequisites** - If there are steps that the customer should complete or tools that need to be downloaded before continuing through the troubleshooting guidance, they should be described in this section.
- **Potential quick fixes** - Sometimes an issue may require long-term work to implement a permanent solution, but can be temporarily taken care of with a quick fix. If this section is added, make sure that information is included about where to look for a permanent solution.
- **Root cause and solutions** - To be able to identify the issue and how to prevent it from happening again, the cause of the issue should be defined if known. Make sure that the H3 headings clearly state the intention of the section. Each section should have a short sentence that describes the steps that are about to be taken.
- **Advanced troubleshooting and data collection** - If the issue requires advanced troubleshooting steps that may require a call to support, list any information or procedures in this section to help the customer prepare for submitting a support ticket.
- **Next steps** - If there are any next steps that should be taken after the issue has been initially resolved, list them in this section.

# Write known issues articles

8/28/2019 • 2 minutes to read

Known issues articles are written to inform the customer that an issue is known by engineering and there is a definite plan to fix it. All entries in the known issues article should be temporary. If a known issue does not have a timeline to be fixed, or is not likely to be fixed in the short term, it is essentially the default behavior. Do not add it to the known issues topic. Instead, document it directly in the specific reference or conceptual topics.

When the known issue is resolved, remove it from the known issues topic, and remove the call-out to the known issue in the applicable conceptual or reference topics. Do not keep the known issue in the topic in perpetuity with a note that the issue has been resolved.

Do not use the known issues topic to announce upcoming breaking changes. Instead, publish a blog post announcing the breaking change.

If you aren't sure whether you should be writing this type of article, see [Choose the right format for a troubleshooting article](#).

Use this [known issues template](#) to get started.

## TOC

If troubleshooting content is to be provided in a TOC, it should be located in a **Troubleshoot** node under the **How-to guides** node. The **Known issues** node is located under the **Troubleshoot** node. For more information, see [TOC structure](#).

## Metadata

Make sure that the `ms.topic` attribute is set to `troubleshooting`. For more information, see the [Metadata overview](#).

## H1 (Headline)

The heading of the known issues article should **Known issues: {service name}** where service name is the name of the feature or service that the issues pertain to.

## Screen shots

Graphics and videos, when appropriate, can be leveraged to explain steps to follow. Visual representation can help customers fix issues faster.

## H2s and H3s

Each known issue should be contained under an H2 heading that concisely states the issue. Under each H2 should be an introduction of the issue. The following H3 headings can be considered:

- **Prerequisites** - If there are steps that the customer should complete or tools that need to be downloaded before continuing through the troubleshooting guidance, they should be described in this section.
- **Troubleshooting steps** - Not all known issues will be correctable, but if so, add this section describing the steps to take to correct the issue.

- **Possible causes** - List known possible causes of the issue.
- **Next steps** - If there are any next steps that should be taken after the issue has been initially resolved, list them in this section.

# Write problem resolution articles

5/10/2021 • 3 minutes to read

Problem resolution articles help customers quickly identify the problem or error that they are having with a service or feature, identify the cause of the problem, and find steps that can be performed to resolve the problem.

- For search efficiency, each problem resolution article should contain only one problem or error per article.
- The audience is customers who are having a problem with a product, service, technology, or scenario.
- The title of the article should specifically contain the message that the customer sees, or include key words from the message that can be used to search.

Use this [problem resolution template](#) to get started. If you aren't sure whether you should be writing this type of article, see [Choose the right format for a troubleshooting article](#).

## TOCs

If troubleshooting content is to be provided in a TOC, it should be located in a **Troubleshoot** node under the **How-to guides** node. Problem resolution articles are located under a **Problem resolution** node under the **Troubleshoot** node. Avoid word wrapping titles in the TOC. For more information, see [TOC structure](#).

## Metadata

Make sure that the `ms.topic` attribute is set to `troubleshooting`. In the value of the `title` attribute, include as much of the message that the customer sees to enable the article to be found through search. Add the customer intent statement as a comment in the last line of the metadata. The customer intent statement format is:

```
#Customer intent: As a < type of user >, I want < what? > so that < why? >.
```

This statement provides a record of the intent of the article for future contributors. For more information, see the [Metadata overview](#).

## H1 (Headline)

The heading of the problem resolution article should use the message that the customer sees if possible, key words from the message, or key words that describe the symptom that the customer is experiencing.

## Introduction

The article should begin with a concise description of the problem to be resolved. The beginning sentences should include as many of the key words from the message or symptoms as possible. The information in the introduction should help the customer decide whether the information applies to the issue that they are encountering.

## Screenshots

If you think that screenshots will offer value in recognizing symptoms or resolving problems, include them. Don't include screenshots that show intuitive UI; these don't add value and take up valuable real estate in your article. For more information, see [Screenshots: How to create, format, and embed in documentation](#).

## H2s and H3s

Don't number H2 and H3 headings. Follow the H2 headings with one or two transitional sentences explaining why the steps need to be performed or how that section contributes to the whole. When possible you should provide the following sections:

- **Prerequisites** - If there are steps that the customer should complete or tools that need to be downloaded before continuing through the troubleshooting guidance, they should be described in this section.
- **Symptoms** - Precisely describe what the customer should be experiencing when encountering the problem. If the title can't contain the complete message, expand on it here. If there is relevant general troubleshooting information available, link to it from here.
- **Cause** - Describe the cause of the symptoms. It is possible that there could be several causes for a problem. List each one as an H3 with **Cause #** where # is a successive number of possible causes.
- **Solution** - List the steps that should be taken to resolve the problem. It is possible that there could be several solutions for a problem. If there are multiple solutions, put them in the order of complexity and provide instructions on how to choose from among them. List each one as an H3 with **Solution #** where # is a successive number of possible solutions. If workaround information is available to temporarily alleviate the symptoms, list them in this section.
- **Next steps** - Include this section if there are 1 -3 concrete, highly relevant next steps the user should take. Delete if there are no next steps. This is not a place for a list of links. If you include links to next steps, make sure to include text to explain why the next steps are relevant or important.

# Write error code list articles

2/29/2020 • 2 minutes to read

Error code lists can be helpful to customers, but only if information is provided to resolve the error. The error codes should be formatted into a table. Columns in the table include **Error code** and **Description**. The description column should include resolution information or link to a [problem resolution article](#).

If you aren't sure whether you should be writing this type of article, see [Choose the right format for a troubleshooting article](#).

## TOC

The error code list is a reference article and should be located under **Reference**. For more information, see [TOC structure](#).

## Metadata

Make sure that the `ms.topic` attribute is set to `troubleshooting`. For more information, see the [Metadata overview](#).

## H1 (Headline)

The heading of the error codes article should be **Error codes: {service name}** where service name is the name of the feature or service that the error codes pertain to.

# Write an Azure Security Baseline article

3/5/2021 • 6 minutes to read

The [Azure Security Benchmark](#) is a collection of articles that document platform-level security recommendations. The V2 benchmark has 11 "Security Controls", each covering a broad area of cloud deployments security and comprising a number of subcontrols. The V2 Security Controls are:

- [Network Security](#)
- [Identity Management](#)
- [Privileged Access](#)
- [Data Protection](#)
- [Asset Management](#)
- [Logging and Threat Detection](#)
- [Incident Response](#)
- [Posture and Vulnerability Management](#)
- [Endpoint Security](#)
- [Backup and Recovery](#)
- [Governance and Strategy](#)

[Security baseline articles](#) are service-level security recommendations for the exact same set of security controls and subcontrols found in the benchmark. They build on the general, platform-level guidance found in the benchmark to give specific, actionable security recommendations to a customer using the service.

For example, here is the [Azure Security Benchmark recommendation for subcontrol NS-4](#), "Protect applications and services from external network attacks":

Protect Azure resources against attacks from external networks, including distributed denial of service (DDoS) Attacks, application specific attacks, and unsolicited and potentially malicious internet traffic. Azure includes native capabilities for this:

- Use Azure Firewall to protect applications and services against potentially malicious traffic from the internet and other external locations.
- Use Web Application Firewall (WAF) capabilities in Azure Application Gateway, Azure Front Door, and Azure Content Delivery Network (CDN) to protect your applications, services, and APIs against application layer attacks.
- Protect your assets against DDoS attacks by enabling DDoS standard protection on your Azure virtual networks.
- Use Azure Security Center to detect misconfiguration risks related to the above.

And here is the [Azure Front Door security baseline recommendation](#) for the same subcontrol, NS-4:

Use Azure PowerShell to create a geo-filtering policy and associate the policy with your existing Azure Front Door frontend host. This geo-filtering policy will block requests from external networks, such as the ones from other countries or regions except United States.

As you see, the Azure Benchmark provides platform-wide information, while the baseline provides guidance specifically for Azure Front Door.

Published baselines only include the sections relevant to the service. Subcontrols which are not applicable are

omitted.

## Process

Azure Security Baselines articles are created from a series of Azure DevOps work items, one per subcontrol. The PM for a service is primarily responsible for providing the content for these work items (but see [Role of the writer](#), below).

An automated process pulls the data from the work items and creates a markdown file, inserting metadata, introductory text, and headers. The article is then staged for review.

Because the content for these articles resides in Azure DevOps, **never modify the markdown file directly**. Any changes made to the markdown of an Azure Security Baseline will be overwritten the next time the article is generated. Instead, update the content in a security baseline article by revising the [ASCB Customer Guidance](#) field, found on "Full details" tab of the service / subcontrol's work item.

The screenshot shows a work item in Azure DevOps with the following details:

- BENCHMARK LINE ITEM 40097**
- Title:** 40097 Azure Front Door: Network Security- NS-4
- Assignee:** Unassigned
- Comments:** 3 comments
- Save:** Save button
- State:** Done
- Reason:** Moved to stat...
- Area:** AzureSecurityControlsBenchmarkContent
- Iteration:** AzureSecurityControlsBenchmarkContent
- Links:** Service Owner Hub, Full Details (circled in red), ASCB AutomatedOr
- Description:** Manage Azure DDoS Protection Standard using the Azure portal: <https://docs.microsoft.com/azure/virtual-network/manage-ddos-protection>
- ASCB Customer Guidelines:** (This field is circled in red)
- Content:** Use Azure PowerShell to create a geo-filtering policy and associate the policy with your existing Azure Front Door frontend host. This geo-filtering policy will block requests from external networks, such as the ones from other countries or regions except United States.
- Tutorial:** Tutorial - How to set up a geo-filtering WAF policy for your Front Door: <https://docs.microsoft.com/azure/frontdoor/front-door-tutorial-geo-filtering>
- Links:** ASCB Documentatio, ASCB Product Gap, ASCB Date Approve, ASCB Date Reviewer

Your changes will appear the next time the article is rebuilt. Preview articles are rebuilt every morning, and you can always reach out to [mbaldwin](#) to request a manual rebuild.

Staged articles have links in each section to the underlying work items, allowing you to easily revise the content at its source:

## NS-4: Protect applications and services from external network attacks

### Tip

To revise the text in this section, update the [underlying Work Item](#).

**Guidance:** Use Azure PowerShell to create a geo-filtering policy and associate the policy with your existing Azure Front Door frontend host. This geo-filtering policy will block requests from external networks, such as the ones from other countries or regions except United States.

- [Tutorial - How to set up a geo-filtering WAF policy for your Front Door](#)

### Azure Security Center monitoring: Vac

To find all of the work items for a specific service, query the [AzureSecurityControlsBenchmarkContent organization](#) with `ASCB Service = <slug-of-service>`. For instance, to search for Key Vault or Azure Database for MySQL work items, you would set `ASCB Service = key-vault` and `mysql`, respectively. You can quickly construct a search by modifying [this query](#).

## Role of the writer

Although the primary responsibility for providing Security Baseline content falls to the PM owners, the content developers have five major roles in the creation of these articles.

### Security content inventory

At the start of the process, writers should conduct a content inventory of existing security-related articles for the service. This will include articles that document service capabilities that enable network security, encryption and data protection, identity and access management (including Managed Identity), monitoring, and logging. The information from these articles will be summarized in the Azure Security Baseline, followed by the URLs.

For instance, in the [Network Security section of HDInsights](#), there are links to <https://docs.microsoft.com/azure/hdinsight/hdinsight-create-virtual-network>, <https://docs.microsoft.com/azure/network-watcher/network-watcher-nsg-flow-logging-portal>, and other content assets that support the security recommendations given.

### Attend the training

When a service is slated for the next batch of Azure Security Baselines, the writer will receive an invitation to an upcoming "Azure Security Benchmark Workshop" (see [Schedule](#), below) These half-day workshops allow PMs and writers to collaborate in populating the Azure DevOps work items with recommendations. In many cases, the bulk of the Azure Security Baseline work can be done during this session.

During the training portion of the event, SMEs on Network Security, Identity and Access Management, and other areas explain the specifics of these Security Controls. This information will serve the writer well, even beyond the creation of the Azure Security Baselines. Attendees will receive credit for training attendance.

The training is a key component of this process. If you are unable to attend the training due to your workload or time constraints, please work with your manager.

### Review and revise the draft Azure Security Baseline

After the work items have been populated, an automated process will create a markdown file and stage it for review. Writers, along with other stakeholders, will review the article and update the underlying work items to correct wording, fix broken or malformed links, improve grammar and clarity, and so forth.

For details on how to update a section, see [Process](#), above.

## Place the Azure Security Baseline in your TOC

By default, newly created Azure Security Baseline articles will be placed in the "Security" node of your TOC's "Concepts" section; if your TOC does not already have a "Concepts > Security" node, one will be created.

Work with the Security Horizontal SME to ensure that the article is placed correctly.

You may wish to use this as an opportunity to collect security-related articles under a single node. You could, for instance, move conceptual articles on Managed Identity or Monitoring under "Concepts > Security".

## Capture security content gaps

As you perform the content inventory of security-related articles and reorganize existing assets under the "Concepts > Security" node of the TOC, you may realize that you have content gaps. Writers should make note of these in their Azure DevOps backlog, with a goal of addressing them before the next version of their Azure Security Baseline is released. (The schedule of Azure Security Baseline updates has yet to be determined, but may eventually settle into a twice-a-year cadence.)

## Maintenance

After the Azure Security Benchmark is updated, baselines will be revised to conform to the newest version. Baselines may also be periodically reviewed and refreshed by the PMs and Azure Benchmark team.

If you notice something amiss in your baseline, or wish to make an out-of-band update, ask [mbaldwin](#) to generate a new preview version, complete with links to the work items. He will also put you in contact with the Benchmark team, to coordinate your effort with the affiliated PMs.

## Schedule

The development schedule for baselines depend on the milestone to which they belong. A full schedule for your baseline will be provided by the Azure Security Benchmark PMs prior to the training session.

## Questions

If you have questions about the security baseline work, use our [Horizontal Content Teams channel](#).

## Next Steps

- The [Azure Security Benchmark site](#) on docs.
- The [Azure Security Benchmark team's wiki](#)

# Write "best practices" content for using your service

5/20/2021 • 5 minutes to read

## NOTE



THIS DOCUMENT IS IN REVIEW AND IS NOT YET SUPPORTED IN THE CONTENT STANDARDS FOR [DOCS.MICROSOFT.COM](#). WE ENCOURAGE YOU TO USE THE GUIDANCE AND PROVIDE FEEDBACK [IN OUR TWO-QUESTION SURVEY](#).

"Best practices" content describes recommendations for working with a service or feature. Customers do not *have* to implement best practices when using your service, but doing so will probably help them have a better experience—either by avoiding common pitfalls, reducing complexity, or otherwise using the service or feature more effectively.

This article contains considerations and recommendations for documenting best practices.

## Determine doc vs. blog post

Best practices content can be part of the documentation for a service, or shared in a blog post by the product team. Here are some considerations to help you determine whether your best practices content is appropriate for docs.microsoft.com.

- **Is this information steady?** Best practices in docs are considered current, active guidance for the service at any time that a customer reads them. If the recommendations may change soon or often, consider a blog post that is more tied to a point in time.
- **Is this information common to many users?** Best practices might belong in docs if they describe common use cases with your feature/service. If they cover more niche operations, consider a blog post directed at that audience.
- **Is this information important to the experience?** Best practices might belong in docs if they are important information that most users should know in order to have a good experience with the feature or service. If these are optional suggestions for advanced users, consider a blog post directed at that audience.

For general guidance on blog content and when to choose blogs vs. documentation, see [Blog process and guidelines](#).

## Scope and organize content

Once you've decided to write your best practices content on docs.microsoft.com, start by thinking about how your best practices articles will be grouped and organized in the context of your service. This will help you define the scope of each article and be clear about the role each one plays in helping customers work with your service.

This section describes three possible strategies for organizing best practices content within your service's TOC. Consider the nature and size of your best practices content, as well as the existing structure of your TOC and content.

STRATEGY	USE WHEN
Best practices section within another concept or how-to article	You may consider grouping best practices for a feature with another piece of content for the feature, if the best practices content for the feature is small. Alternatively, if there are many best practices to explain, consider pulling out best practices into their own article(s) in one of the other strategies below, to keep from overwhelming the original intent of the concept or how-to article.
Single best practices article for the service	Use when your service is tightly-scoped and the best practices apply to the service as a whole <i>OR</i> Your total amount of best practices content is small (1-2 best practices per feature area)
Multiple best practices articles, grouped with similar topics in TOC	Use when your best practices span several feature areas or parts of the service <i>AND</i> You have more than 1-2 best practices for each feature area <i>AND</i> Your TOC lends itself to grouping by feature or topical area of the service
Multiple best practices articles, grouped together in Best Practices category in TOC	Use when your best practices span several feature areas or parts of the service <i>AND</i> You have more than 1-2 best practices for each feature area <i>AND</i> Your TOC is not topically grouped

## Create content

This section deals with writing a best practices article. The content of the article will vary depending on your service and the amount and nature of the content, but you should consider the following general suggestions.

### Choose article type

First, decide what [article type](#) is most appropriate for your best practices content. These considerations might help:

- A [concept](#) article is often a good choice to convey information about a process or idea, without making assumptions about the customer's environment and what specific implementation will work best for them. If your best practice is an idea, consider using a concept article to describe how it works and provide general guidance that a customer can adapt to their own specific situation.
- A [how-to](#) article may be appropriate if your best practice requires specific and detailed steps to implement, and the steps will be similar for almost all customers using this recommendation.

### Set expectations in the intro

In the introduction for your article, be sure to set context for readers on what "best practices" means in this case. Help them understand that these recommendations aren't required and may not be appropriate for every user, and understand whether a recommendation is right for them. Be clear about the potential benefit of following the best practices, as well as any potential tradeoffs that readers should understand before implementing.

### Organize content within the article

There are many ways to organize the content within a best practices article. Here are some suggestions:

- **Group by topic:** A straightforward way to group your best practices is by topic or feature. If your best

practices span several topic areas or features within your service, you may want to use this organization strategy.

- **Group by role:** If you have clearly defined roles for your users, consider grouping recommendations by the role that will need them.
- **Group by tool/service:** If different tooling options are important for your service, consider grouping recommendations by tool to which they apply. For example, if you have many best practices around Docker configuration, consider grouping these together so developers can do all the Docker configuration they want to do at once.

### Reference existing best practices content

As you create service-specific best practices content, consider referencing other best-practices content that might affect your readers more generally, like **best practices for Azure** or **.NET developers**, or **best practices for your topic area within Azure**. Reviewing relevant best practices articles that already exist can help you avoid repeating content by linking to them where appropriate.

For example, if you are writing about a service in Azure IoT, you might want to review the following articles and consider linking to them if they apply to your service-specific developers:

- [Secure development best practices on Azure](#) (applies to Azure developers)
- [Security recommendations for Azure Internet of Things \(IoT\) deployment](#) (applies to Azure IoT developers)
- [Azure IoT reference architecture](#) (applies to Azure IoT developers)

## Plan for longevity

It's important to have a plan for maintaining best practices content, since it may change as new features or ways of working with your service are introduced. Best practices in docs should always reflect current, up-to-date recommendations.

Consider having a PM partner for each topic area that can periodically review the content to make sure it stays current and relevant.

# Legal content

3/5/2021 • 2 minutes to read

Documentation in large repos such as azure-docs-pr and sql-docs-pr is subject to machine translation, public feedback, and automatic merging of internal contributions if the changes are within the [PR Merger threshold](#). This can pose a risk to the company if legal or policy documents are incorrectly translated or unintentionally modified.

To mitigate this risk, we isolate all content with potential legal implications in a private, locked-down repo: [MicrosoftDocs/DocsLegal-pr](#). This content must receive applicable CELA reviews and follow a special [localization process](#). Each product team is responsible for maintaining the integrity of content in their subfolder of this repo.

## NOTE

Before you add content to the DocsLegal-pr repo, review existing legal and licensing webpages to see if they apply to your scenario. If you determine you need additional documentation, consult your [CELA representative](#) to confirm the need.

## File and folder requirements

Be aware of these requirements before you contribute to the DocsLegal-pr repo:

- Individual product teams are responsible for completing necessary CELA reviews and updates.
- Individual product teams are responsible for confirming localization needs with CELA, and then coordinating translations and publication to localized repos. This includes funding those translations. See [Localization of legal content](#) for detailed instructions.
- Each product or service must create and use a subfolder in the repo.
- Filenames must be lowercase with hyphen separators.
- All files must include the [required metadata](#) for Developer Relations. The DocsLegal-pr repo uses the [Developer Relations rule set](#) to enforce these values.
  - If your file is not product or service specific, use `ms.prod: legal`
  - All files should use the generic topic type `ms.topic: article`

We recommend that all files use an additional metadata value: `layout: ContentPage`. This creates a flat-page experience with no TOC. The exception to this is groups who want to create a [contextual TOC](#) to make legal articles appear within the TOC of a product or service guide. *Do not* create a regular TOC within the DocsLegal-pr repo: the root TOC and breadcrumbs are out of context and will confuse readers.

## Publication process

Use the following steps to contribute source content to DocsLegal-pr, which publishes to the en-us docs.microsoft.com site.

1. Draft your Markdown topic in a [forked copy](#) of [MicrosoftDocs/DocsLegal-pr](#). Refer to the [requirements](#) above.
2. Create a pull request (PR) to the master branch of the upstream repo.
3. Review the staged pages from the [Preview URL](#) links in the PR and ensure the content is equal to the CELA-approved document. Complete any other reviews your team requires. For example, some product

groups have their CELA representatives review staged content.

4. Confirm that the PR built successfully, address any validation errors or warnings, then add the `#sign-off` comment if you are ready for the content to go live.
  - If you require special (timed) or rushed publication, do not `#sign-off`. Instead, add a comment to the PR and explain the request.
5. Content Production Services (CPS) monitors the repo for new PRs, and will comment if the PR is not ready to be merged. You must address these comments.
6. CPS publishes all merged PRs to the live site once a day.

After your content goes live in the main (en-us) repo, you can start to coordinate your [localization requests](#), if needed. It is important that the English content is published **before** you move forward with updates to translated content.

# Legal guidelines

6/1/2021 • 2 minutes to read

Here's a summary of legal guidelines for all articles on docs.microsoft.com (Docs). You can find complete guidelines here: [CELA Web \(Corporate, External, and Legal Affairs\)](#).

## Fictitious names

Use fictitious company names, URLs, and email addresses in all content.

### Company names

- **Correct:** Fabrikam, Inc. (Other examples: Contoso, Ltd., Fourth Coffee, Bellows College)
- **Not correct:** Walmart

### URLs

- **Correct:** <http://www.fabrikam.com/>
- **Not correct:** <http://www.apple.com/>

### Email addresses

- **Correct:** someone@example.com
- **Not correct:** abraham@lincoln.com

### Street addresses

- **Correct:** 4567 Main St, Buffalo, NY 98052 (sequential numbers, incorrect zip codes)
- **Not correct:** 1600 Pennsylvania Avenue NW, Washington, DC 20500

For more information, see [Approved fictitious names & guidelines](#)

## Talking about the future

Don't discuss future functionality in technical documentation. Technical docs tell users how technologies work, not how they might work.

### From Legal:

Documentation is part of the product. Users can claim to have been influenced to make a purchase based on a promise we make in the docs that we ultimately don't keep. Announcements about future functionality are more appropriate for blogs and marketing.

- **Correct:** The following Azure service currently has built-in publisher support for event grid: Event Hubs.
- **Not correct:** The following Azure service currently has built-in publisher support for event grid: Event Hubs. Other Azure services will be added this year.

## Linking to third-party sites

Sometimes it's necessary to link to third-party sites because MS technologies interoperate with many non-MS technologies. The [docs.microsoft.com - Terms of Use](#) on Docs covers these links. *But include them only when the content is a trusted source and when reproducing the content on Docs is impractical.* Follow these guidelines:

- **Consent:** Make sure you have consent from the third-party if necessary. You can determine whether it's necessary by reviewing their terms of use. If they say their content can't be used without their consent or for commercial purposes, then contact them. Explain your request and, if they agree, get their consent in an email. It's courteous and a good idea to get their consent under any circumstances.
- **Sign off:** Make sure PMs or other SMEs sign off on the third-party info. It should be trusted content.
- **Freshness:** Review the links as a regular part of content freshness.
- **User experience:** Jumping to another site is a bad user experience, so let users know they're leaving Docs, as in this example:

**Prerequisites:** To complete this tutorial, you need Android Developer Tools, which you'll find on the [Android Developer site](#). The tools include the Android Studio IDE and the latest Android platform.

## Plagiarism

Don't copy someone else's content. You discredit yourself and Microsoft, and you can get in trouble, possibly publicly.

### What to do:

- Rewrite, don't copy.
- Give credit and link to the source.

## GDPR

Technical documentation can't use the terms "General Data Protection Regulation," or "GDPR" outside the context of the CELA-approved includes listed in this [article](#).

### NOTE

Using the terms "General Data Protection Regulation" or "GDPR" outside the context of the [CELA-approved includes](#) requires approval from CELA before the PR can be signed off.

## Next steps

- [CELA Web \(Corporate, External, and Legal Affairs\)](#) for comprehensive legal information for Microsoft employees.
- Consult your legal contact with any questions about the guidelines.

# docs.microsoft.com/samples Basics

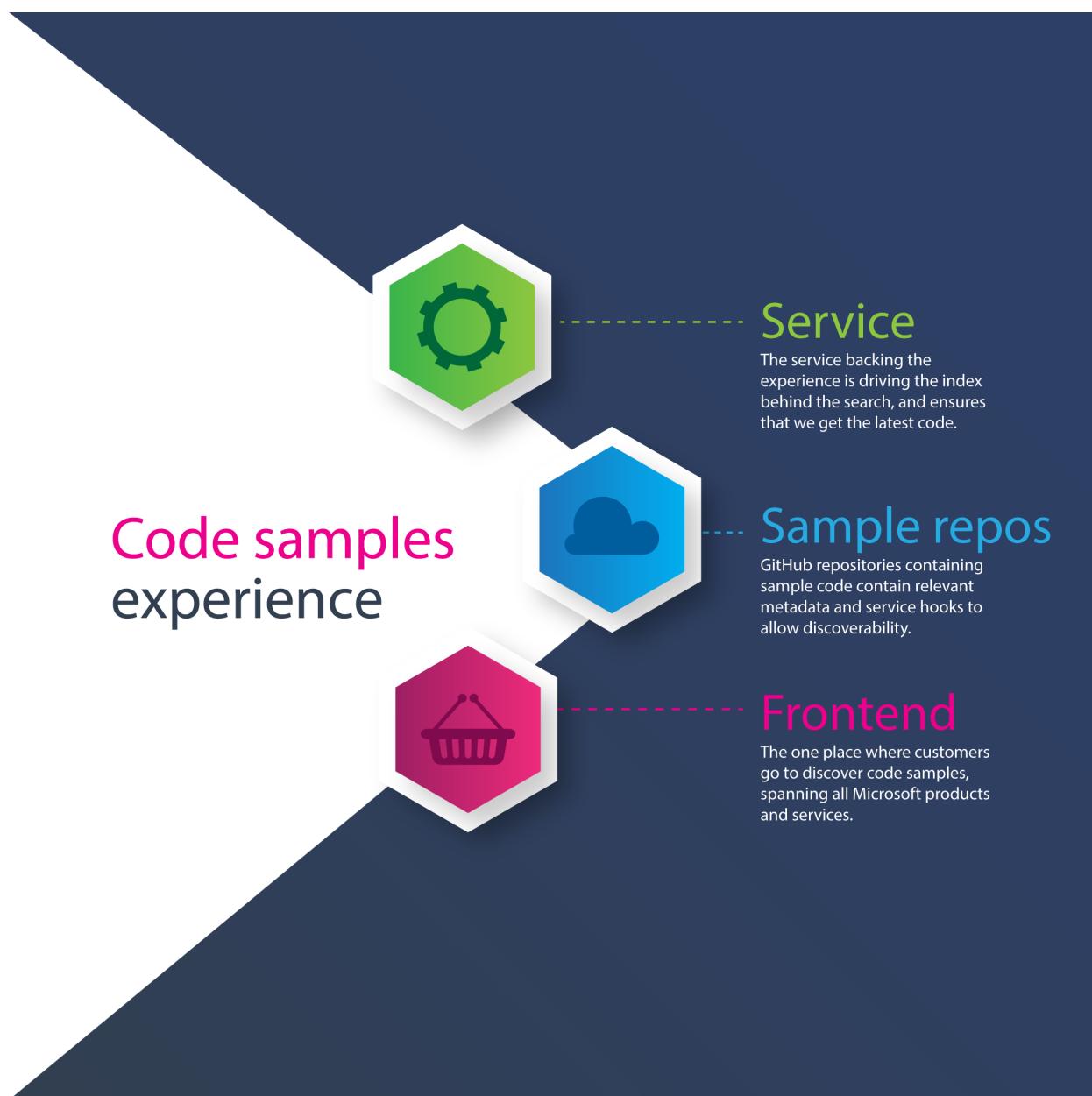
11/2/2020 • 3 minutes to read

Our team built [a centralized experience for code sample discovery](#), with the following goals:

- Create *one* place where any developer can find samples for *any* Microsoft tool, product, or service.
- Power the experience with the flexibility of source control and community contributions made on GitHub.
- Make it easy for partners to be a part of the new code discovery and distribution site.

The guidance in this section will show you how you can bring your samples to the experience and make sure that your code shines and addresses customer needs.

## Building blocks



### Service

The backend services are crawling onboarded code sample repositories hosted on GitHub, ensuring that we document the latest version of the code. These services also index the content that is used to populate the

[samples browser home page](#). Last but not least, these services also are responsible for producing downloadable ZIP files, that customers can use instead of cloning the repository.

The main indexing service is connected to GitHub repositories through [a webhook](#), which tracks repository changes and processes updates whenever those happen. The webhook does not monitor pull requests - only commits to branches within the repository.

In most cases, partner teams and external customers will never need to know the inner workings of any of the backend services service. We are planning on building an online dashboard in the [Docs Portal](#), where it will be possible to see the onboarding status and any issues. If you encounter any issues with your samples, refer to the [troubleshooting guide](#).

## Sample code repositories

Individual repositories that contain code samples are hosted on GitHub, in one of the [Microsoft-sponsored organizations](#). One repository can contain one or many code samples.

Each sample repository that is onboarded to the code samples experience has:

- A `README.md` file for each sample or group of samples exposed to external customers ([learn about our guidelines for the `README.md` file](#))
- Metadata included in the `README.md` file for each sample or group of samples that is exposed to external customers ([learn about authoring metadata](#)).
- Webhook that tracks changes in the repository and is connecting the repository to the backend service.
- Our service account ([VSC-Service-Account](#)) added as a collaborator with **Write** permissions to create new release ZIPS.

### IMPORTANT

We do not support onboarding code samples from personal repositories or from organizations that are not approved by the [Open Source Programs Office](#) at this time.

## Customer-facing experience (front-end)

The [main page for all samples](#) is called the **samples browser**. It is the one place where a customer can find the relevant code samples by searching or applying product and/or programming language filters.

The screenshot shows the Microsoft Docs website with the URL <https://review.docs.microsoft.com/en-us/new-hope/styleguide/template-samples.html?branch=master>. The page title is "Code samples". A search bar at the top right contains the placeholder "Search". Below the search bar is a "Refine" section with dropdown menus for "Platforms" and "Products", both currently set to "57 matches". A "Keyword filter" input field is also present. The main content area displays three code sample cards:

- Create serverless applications** (NEW): 00/00/2018. Description: Learn how to leverage functions to execute server-side logic and build serverless architectures. Tags: Developer, Intermediate, Azure, +7.
- Build a serverless web app** (COMPLETED): 00/00/2018. Description: Deploy a simple web application that uploads images via an HTML interface and utilizes a serverless back end for automatic tagging. Tags: Developer, Intermediate, Azure, +7.
- Provision an Azure SQL Data**: 00/00/2018. Description: Manipulate SQL data easily after learning how to do this. Tags: Developer, Intermediate, Azure, +7.

When a customer discovers a specific code sample, they are able to open the page that we generate from the `README.md` file for the sample or group of samples on GitHub:

The screenshot shows the Microsoft Docs website with the URL <https://review.docs.microsoft.com/en-us/new-hope/styleguide/template-sample.h>. The page title is "Azure IoT Samples for Csharp (.NET)". The page header includes the Microsoft logo, navigation links for "Docs", "Documentation", "Learn", "API Browsers", and "Code samples", and a search bar. The main content area has a heading "Azure IoT Samples for Csharp (.NET)" and a timestamp "00/00/2018 • Contributors". It features three download buttons: "Deploy to Azure", "Browse code", and "Download ZIP". A brief description states: "Azure-iot-samples-csharp provides a set of easy-to-understand, continuously-tested samples for connecting to Azure IoT Hub via Azure/azure-iot-sdk-csharp.". The "Prerequisites" section lists ".NET Core SDK 2.1.0 or greater on your development machine. You can download the .NET Core SDK for multiple platforms from [.NET](#). You can verify the current version of C# on your development machine using 'dotnet --version'." The "Resources" section links to "azure-iot-sdk-csharp", "Azure IoT Hub Documentation", and "Get-started". The "This sample demonstrates" section is partially visible at the bottom.

Within this page, you can:

- Read the sample details, imported from the sample `README.md` file.
- Deploy a sample to Azure (*if an Azure Resource Manager template is available*).
- Browse the code on GitHub.
- Download the specific sample (*by downloading the ZIP and not the entire repository*).

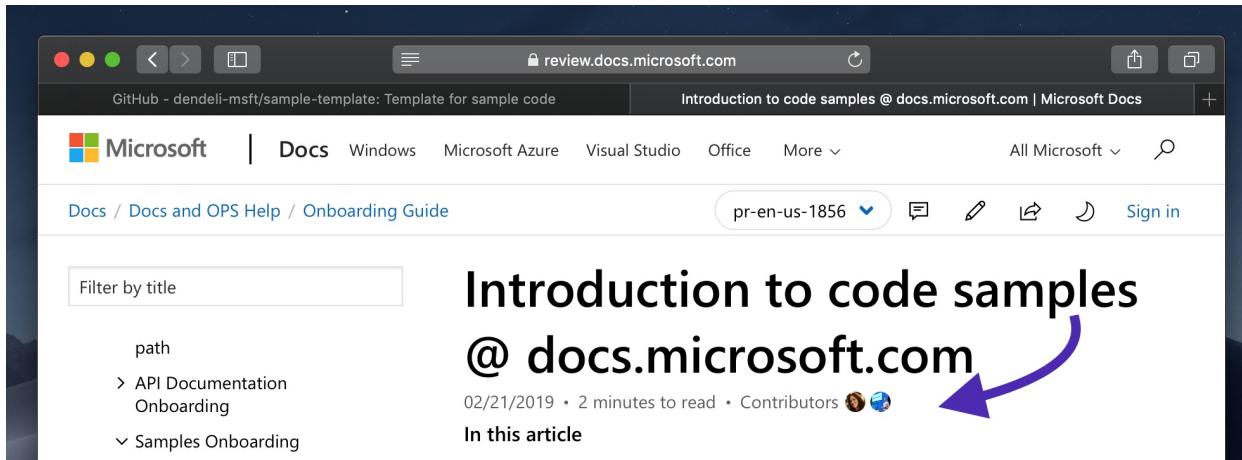
# Onboarding

To get started, read our [onboarding guide](#).

## FAQ

### How can customers outside Microsoft contribute?

At this time, the samples browser is designed to support samples that are officially supported by Microsoft. Any customer can add their GitHub contributions to the sample in the existing code repository on GitHub, and we will properly credit them the same way we do today in our documentation - with the help of the **Contributors** section in the header:



The current workflow also ensures that customers sign the right CLA forms, and are giving us the proper rights to reuse the redistribute the code.

# Governance for code samples at Microsoft

6/8/2021 • 9 minutes to read

This document outlines the governance for code samples delivered by Microsoft to the public. Patterns and practices documented here offer a set of recommendations that will ensure that code samples set you up for success, and are seen as a key part of the developer journey.

## Categories of samples

Samples are non-production code, used by customers for learning and experimentation purposes. A sample can fall into one of the following categories:

- *Code snippet* - a short piece of code, that shows how to use a very specific API or a set of APIs. It usually requires the user to have some familiarity with the development environment, and the ability to create a new project in which to test the code snippet. A snippet is not always a piece of standalone testable code.
- *End-to-end sample* - a full-blown project that the user can download, open, build, and deploy with minimal changes.

This guidance specifically addresses the needs of *end-to-end samples*. The document below should be treated as a de-facto checklist for your sample code release. Best practices for code snippets are managed in the [Code in Docs article in the contributor guide](#).

## Guidelines

### Content

- Checklist item:** *Ensure your code is well-written and is useful to external developers.*

Sample code should be easy to understand and use, but complex enough to be useful to someone who wants to implement a specific scenario, or learn how to work with a specific API.

Developers are likely to likely copy and paste the code in both development and production scenarios. If you are aware of any of the following problems, be sure to put a disclaimer in the `README.md` file in the code sample project.

- The code is not secure (for example, connection strings are included in the code).
- The code is grossly inefficient (for example, loops that run unnecessarily because it uses the wrong breakout points).
- The code does not run (for example, there are build errors).
- The code may cause a problem if used in a production scenario.

Code samples should be self-contained and in most cases not require the user to clone/download content from other locations (other than dependent libraries). That is not always possible to avoid, so if you find yourself in a situation where you need to connect a number of dependencies, always include helper scripts and detailed instructions that get the user started as soon as possible.

Do not include code that is not relevant to the scenario presented within the sample itself. For example, if you are showing how to build for the Xbox Live achievements service, try to scope it to just that and avoid including code that shows how to interact with in-game chat.

### Legal

- Checklist item:** *Ensure your code is compliant with legal requirements for shipping open source software at Microsoft.*

You can release code samples without formal legal approval, if the following conditions are met (see [latest release policy](#) for an up-to-date list):

- You exercise good judgment and only release open source in the best interest of Microsoft.
- You release under the MIT license.
- The code is less than 5,000 lines, including any automatically generated code.
- The release does not distribute any third-party code or content. Dependencies installed by the consumer (for example, via a package manager) are not considered distributions.

If any of the above do not apply to your code sample, you will need to ensure that you have CELA approval to publish the code sample.

## Hosting

- Checklist item:** *Ensure your code is stored in a Microsoft-sponsored GitHub repository.*

All samples that Microsoft distributes should be hosted on [GitHub](#), in one of the Microsoft-sponsored organizations, such as [Azure](#) or [Azure-Samples](#), in a public repository. The full list of organizations that fall into this category is maintained by the **Open Source Programs Office (OSPO)** and are accessible via the [Open Source Portal](#).

### WARNING

Code samples **should not be stored in personal repositories**. Any code samples stored in personal repositories will not be shown or discoverable in the [samples browser](#) - we will not be able to onboard them due to legal and compliance reasons.

### WARNING

We do not support onboarding samples from Azure Repos or from private GitHub repositories.

## Licensing

- Checklist item:** *Ensure your code has a license that will not put Microsoft or its employees at risk.*

All samples should follow a licensing model that allows distribution, modification, and adaptation. The recommended license is the [MIT license](#).

When the sample is hosted on GitHub, it's [possible to specify what the license should be](#), and an automated process will create a [LICENSE](#) file for you.

### NOTE

If your sample code is using a license different than MIT, you will need to undergo [legal review](#) before the sample can be released to the public.

## Structure

- Checklist item:** *Ensure your code is well-structured within the repository and provides enough context to get started.*

Each sample should start with the common set of templates to ensure consistency across all projects. At a minimum it should contain the following entities:

COMPONENT	DESCRIPTION
.github folder	Containing templates for creating <i>Issues</i> and <i>Pull Requests</i> .
README .md	<p>See <a href="#">Sample README Template</a> to learn about the standard sample template.</p> <p>Specifies the project name, description, features, requirements, and information to run the sample. It also includes information on dependencies or pre-requisites, along with guidelines on how to build and deploy. Multiple <code>README .md</code> files can be created for different samples within the same repository. Each <code>README .md</code> file in the repository, that covers an individual sample or set of samples should have the required metadata integrated to allow discovery in the <a href="#">samples browser</a>.</p> <p>Ideally, the <code>README .md</code> file also contains rich media, such as static images or GIFs, that showcase the final output to the user, if applicable.</p>
LICENSE	Code sample license. See information about <a href="#">licensing</a> previously explained in this document.
CONTRIBUTING .md	Share details on how to contribute to the sample as well as the code of conduct that contributors must follow.
CHANGELOG .md	Describe the feature changes, bug fixes, and other notable modifications that were made in each version of the sample.

It is also advised for the code within the sample to have comments, to make it easier for users to understand the inner workings of the project.

Make sure to keep your repositories structured in a way that allows customers to navigate them purely by following the folder names. Having descriptive file and folder names also makes maintenance easier.

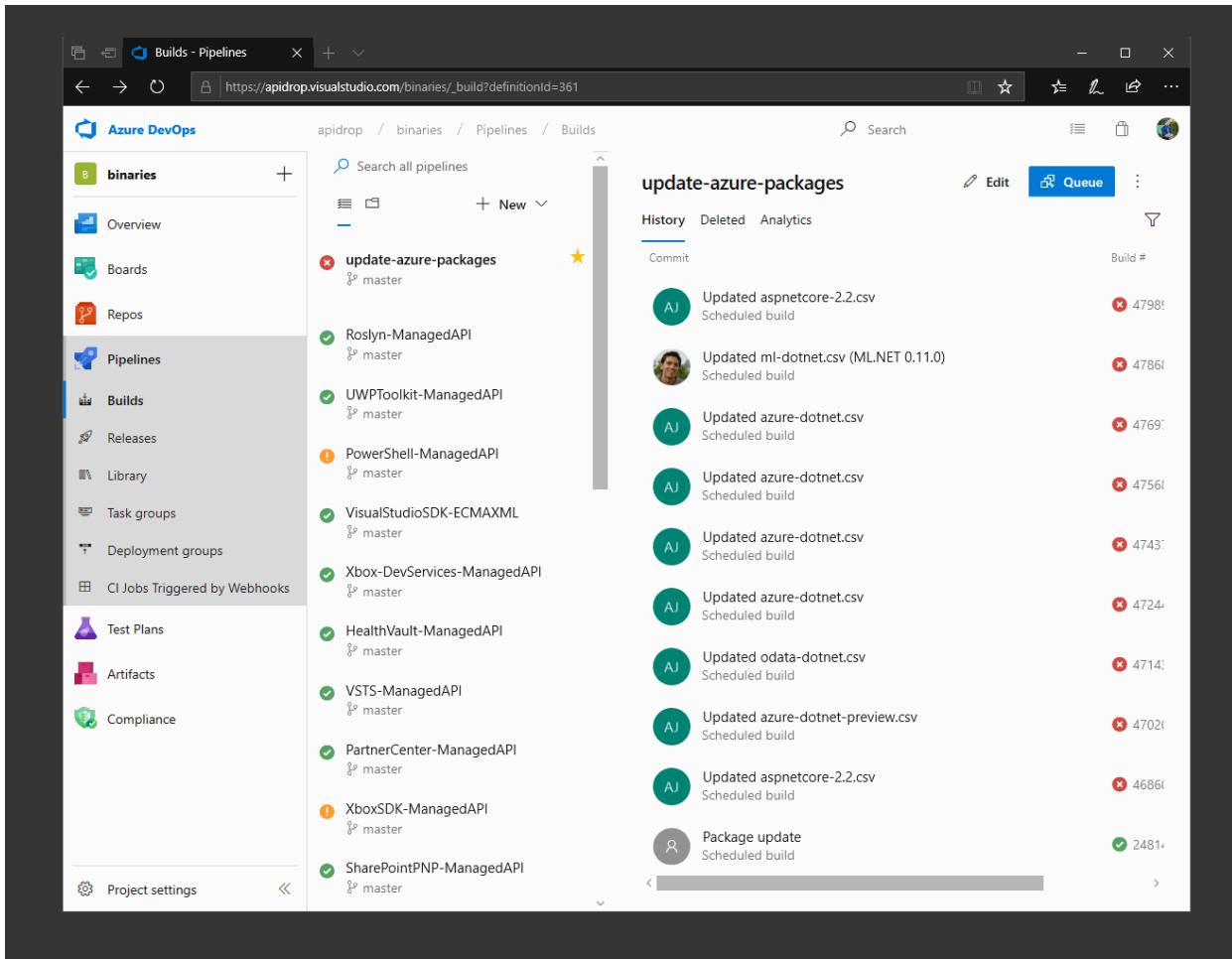
#### IMPORTANT

Do not check in binaries and non-code files, as this greatly increases the size of the repository that has to be cloned in order to access the sample. It also introduces security and licensing risks.

#### Testing and validation

**Checklist item:** *Ensure your code is regularly tested and validated.*

All samples should be tested and validated, ensuring that our customers will always get the most reliable and up-to-date code. Your team is responsible for ensuring that you connect the sample to a continuous integration system, and are presenting the build status in `README .md`. You can use [Azure Pipelines](#) or [GitHub Actions](#) and many of the built-in templates to set up your testing environment.



We will start sending out broader reports in the near future that will bring up repositories that do not have a CI job configured.

## Deployment

- Checklist item:** Ensure your code can be deployed to Azure easily, if it deals with Azure resources.

If the sample requires the user to deploy resources to Azure, it should provide clear guidelines on how to do that, along with an [Azure Resource Manager template](#), where applicable.

If you are providing ARM templates to customers, make sure that you are following the [existing best practices](#) and [test them](#).

## Samples browser metadata

- Checklist item:** Ensure your code is discoverable in the samples browser.

To make sure that the sample can be discovered by customers in the [samples browser](#) the `README.md` file needs to include metadata that allows our service to index it. You can read more about the required metadata as well as repository configuration changes in the [onboarding document](#).

Metadata needs to be in the form of [YAML front-matter](#) in the `README.md` file representing the sample or group of samples, depending on how your team expects the customers to get the code.

## Security

- Checklist item:** Ensure your code is secure and does not depend on vulnerable components.

As a rule of thumb, your code should be showcasing the best security practices and should not include any private information (e.g. *database connection credentials, API keys*). Assume that whatever you write in your sample code will be used as-is in production services.

You can learn more about open-source security best practices and guidelines in the [Open source security](#)

document.

## Maintenance

Sample maintenance is the responsibility of the original sample creator and/or team assigned to the project. The docs.microsoft.com team offers governance and discoverability features, but does not diagnose code-specific issues, implement and maintain testing infrastructure, or define the lifecycle for the code.

When the code is no longer maintained, the team should [archive the repository](#). This signals to customers that the code is no longer updated, and no reported issues will be addressed. Whoever is maintaining the sample code can determine the appropriate time and strategy for archiving the repository. However, when we are informed by the product or content teams that the code is no longer applicable to external customers, we will remove it from the samples browser.

## Community contributions

### License agreement

All repositories should be public, and support community contributions from external customers. Before accepting community contributions to the sample, it's important to require a Contributor License Agreement (CLA) from any non-Microsoft users.

A CLA service integrates with GitHub pull requests to provide information about the contribution status of a pull request, while also offering an opportunity for new community members to get connected to sign the necessary legal document.

A CLA bot is installed automatically in all repositories that are located in Microsoft-sponsored organizations.

### Adding code

Customers can add code to sample repositories with the help of pull requests. It is the responsibility of the team or person managing the code sample to determine whether the contribution adds value and is relevant to the scenarios handled by the product team.

A user can submit entirely new code samples to an existing repository. The guidelines above would still apply to any community-sourced code.

## Referencing in documentation

You can reference code samples in documentation hosted on docs.microsoft.com. Make sure that whenever you need to provide a link to a sample, you link directly to the page for it in the [samples browser](#) and not GitHub. This allows us to provide custom experiences for our users, such as one-click deployments to Visual Studio Online and direct downloads.

You should avoid including links to separate samples in the table of contents. Instead, make sure to make it clear to your users where they can find code samples as they read the documentation for your product.

## Scripts

When creating samples that are **scripts** and are not end-to-end solutions, you should consider storing them in one of the following locations:

SCRIPT TYPE	REPO
Azure CLI	<a href="https://github.com/azure/azure-docs-cli-python-samples">https://github.com/azure/azure-docs-cli-python-samples</a> (Store the scripts in a folder that's specific to your service.)

SCRIPT TYPE	REPO
Azure PowerShell	<a href="https://github.com/azure/azure-docs-powershell-samples">https://github.com/azure/azure-docs-powershell-samples</a> (Store the scripts in a folder that's specific to your service.)
Resource Manager templates	We encourage teams to use <a href="https://github.com/Azure/azure-quickstart-templates">https://github.com/Azure/azure-quickstart-templates</a> . For more information, see that repo's Contribution Guide. However, the repo that your engineering team uses may be different.

Every script should be treated like a separate sample, that should have its own `README.md` markdown, that outlines what the script does, and contain the required metadata, outlined in the [onboarding document](#).

In your content, do not point users to GitHub to get the scripts, but rather to their respective pages on [docs.microsoft.com/samples](https://docs.microsoft.com/samples).

## Next steps

- [Create and publish sample code](#)

# Sample README Template

11/2/2020 • 11 minutes to read

This document outlines the requirements for the `README.md` file for code samples developed by Microsoft and hosted on GitHub. A `README.md` file presents important information that allows the customer to get started with a service or application. It is an opportunity to provide as much context to set the customer up for success. It is recommended that `README.md` writers read [The Art of the README](#) before proceeding with the guidelines below.

## General guidelines

- A `README.md` should use simple language. Avoid acronyms, overly technical terms, and slang.
- A `README.md` is consumer-first. Its primary target audience is customers outside Microsoft who might have zero context on the target service or product.
- A `README.md` that is long doesn't mean that it's of a higher quality. Brevity is important - for more extensive documentation, point people to dedicated resources (for example, [docs.microsoft.com](#)).
- A successful `README.md` file is one that enables customers to get the code and run it without having a deep understanding of how and why it works.
- A `README.md` that follows a predictable format is easier to parse. It is important to ensure that the file is kept as consistent as possible with the examples below.
- Make sure to use absolute links instead of relative ones within your `README.md` file. This guideline ensures that your page renders successfully in the [samples browser](#). It also helps prevent broken links that result in a 404 error.

## Media

A good `README.md` file includes relevant media, such as images, screenshots, and diagrams where applicable. This content format helps customers have a better grasp of concepts covered in the sample, and provides a richer presentation within the [docs.microsoft.com/samples](#) experience.

For example, [The Weather Experience](#) sample gives a very clear idea as to what the developer will be building.

### NOTE

Where possible, you can add interactivity by including animated GIFs. Use [Kap](#) on macOS or [ScreenToGIF](#) on Windows to capture your screen.

When including images in your sample, make sure you create the standard folder called `media` and put pictures and videos there. This consistency makes it easy for others to figure out where the content is stored.

When including images in your sample for use in your `README.md` file, be sure to use the standard Markdown format -- do not use HTML, or you will have broken links when the sample is displayed in the [samples browser](#).

## Recommended structure

Every `README.md` should be a good standalone entity - both on GitHub as well as a rendered page on [docs.microsoft.com/samples](#). If a customer finds the file on GitHub, they should be able to do everything to get started with just that file as the context. If they find the sample in the [samples browser](#), they should be just as

successful.

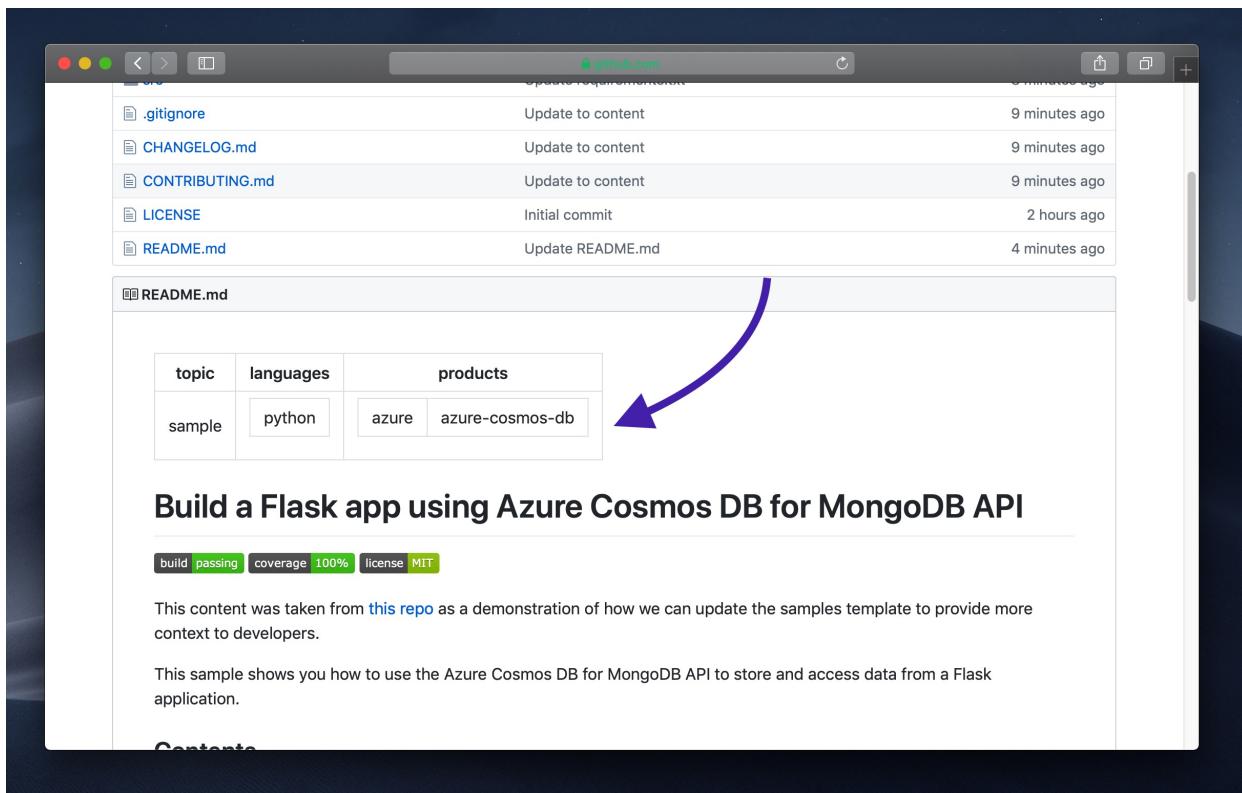
#### NOTE

You can see an example of an implementation of the guidelines below in the [docs-product/sample-template](#) repository.

## Metadata

Every sample that Microsoft ships should be discoverable in the [samples browser](#). To ensure this, every sample needs to have metadata integrated in relevant `README.md` files, giving the indexing tooling a way to properly categorize them in the big set of samples we document.

The metadata is included in `README.md` files as [YAML front-matter](#). You can read more about the required metadata values in the [Onboarding Samples](#) document.



#### NOTE

It is important that in addition to setting up `README.md` metadata, you also configure the repository with the appropriate webhook and service account access - this allows our indexing service to track code sample changes and update those in the [samples browser](#).

When setting up the metadata, make sure that you refer to [Taxonomies for Docs](#) to learn more about the accepted values for product and programming language fields. The slugs used in the front-matter will represent the lists of products and programming languages that your sample uses - this information will make it easier for customers to discover the samples that are most relevant for them.

Some important considerations:

- Try to specify a `urlFragment` where possible ([see example](#)). This ensures that the generated URL stub for the sample page on [docs.microsoft.com/samples](#) is immutable. Otherwise, we will generate the link from the name of the sample, and if that changes, the link will change as well. No automatic redirection will be done in that case, which can result in broken links if others have linked to the sample from other

locations.

- Try to specify an explicit `description` value. It should be no longer than 150 characters, and should give a quick overview of what the sample is about. In absence of this metadata, we will attempt to read the first 150 characters following the sample name/H1 header - that can be imprecise and may result in unexpected side-effects, such as using the build badge Markup as the sample description.
- If the product or language value that you are trying to use is not available in the [taxonomy document](#), contact [Rachel Price](#) and [Dana Bublitz](#) with the request. You can read more about this process in the [Requesting new metadata values](#) section of our onboarding document.
- If the title of your sample is different from what you want to show on the card rendered in the [samples browser](#), specify a `name` metadata value.

## Title

The title of the sample immediately follows the metadata. It is written as an H1 heading (# in Markdown). This title should be a short description of the sample, and should use sentence-style capitalization (see: [Microsoft Style Guide](#)).

### IMPORTANT

Use standard Markdown conventions. Specifically, use # for header differentiation, and not any other structures (e.g. =====). If you don't follow the convention, the sample indexer will not work correctly when it comes to reading in your code

As a best practice, avoid including images, links, or any Markdown markup within the header text. When such content is included, it will be rendered as plain-text in the [samples browser](#).

### Example of a title

```
# Build a Flask app using Azure Cosmos DB for MongoDB API
```

## Badges

### NOTE

This section is optional. However, you should try and ensure that you are building your sample using some continuous integration workflow. For more information, see [Building your first pipeline](#). Make it clear to customers as to whether the code is currently working.

Badges that show build, test and license information are very important to customers. In many interviews, those were called out as some of the ways in which developers assess whether they should even bother getting the sample and running it. After all, a failing build does not inspire confidence. This situation can get even worse when it fails after the customer downloads and opens the sample.

Badges should be unobtrusive and not take significant horizontal and vertical space. They should be stacked vertically if more than one badge is used. At five badges, it becomes visual clutter that is hard to parse at a glance, so don't include more than four badges.

The screenshot shows a GitHub repository page titled "Build a Flask app using Azure Cosmos DB for MongoDB API". At the top, there are navigation links for "Onboarding", "CPS", "API Docs Team", "Samples Board", "Samples Query", ".NET SDP Board", ".NET to SDP Query", "DevRel - Spec Gallery", and "PM@Microsoft". Below the header, there are four tabs: "sample", "python", "azure", and "azure-cosmos-db". A purple arrow points from the "sample" tab towards the badge area.

Below the tabs, there are three badges: "build passing", "coverage 100%", and "license MIT".

A note states: "This content was taken from [this repo](#) as a demonstration of how we can update the samples template to provide more context to developers."

The main content area starts with a heading "Contents" followed by a table:

File/folder	Description
src	Sample source code.
.gitignore	Define what to ignore at commit time.
CHANGELOG.md	List of changes to the sample.
CONTRIBUTING.md	Guidelines for contributing to the sample.
README.md	This README file.
LICENSE	The license for the sample.

At the bottom of the page, there is a section titled "Prerequisites".

Badges can be different, depending on the team, build system, and code coverage tools. It's recommended to use [shields.io](#) as the service for badge images, or you can use badges provided to you by the build system you are using (such as [Azure Pipelines](#)).

Badges should link to individual continuous integration jobs that apply to the project. If the continuous integration jobs are not public (for example, they are in a private Azure Pipelines instance), build badges should be included, but not linked to anything as external customers do not have access to our internal build jobs.

#### Example of badges

```
![Build passing](https://img.shields.io/badge/build-passing-brightgreen.svg) ![Code coverage]
(https://img.shields.io/badge/coverage-100%25-brightgreen.svg) !![License]
(https://img.shields.io/badge/license-MIT-green.svg)
```

#### Description

Always try to include an overview of the sample, communicating to the customer what they will get out of trying the code. The description should be at most two paragraphs, and should cover some of the fundamental items in the sample. This is not the place to write extensive documentation (more on that later) - this is the elevator pitch that allows the customer to understand whether they should use the sample or not.

#### IMPORTANT

The section should not have a heading.

topic languages products

sample python azure azure-cosmos-db

## Build a Flask app using Azure Cosmos DB for MongoDB API

build passing coverage 100% license MIT

This content was taken from [this repo](#) as a demonstration of how we can update the samples template to provide more context to developers.

This sample shows you how to use the Azure Cosmos DB for MongoDB API to store and access data from a Flask application.

### Contents

File/folder	Description
src	Sample source code.
.gitignore	Define what to ignore at commit time.
CHANGELOG.md	List of changes to the sample.
CONTRIBUTING.md	Guidelines for contributing to the sample.
README.md	This README file.

The description should be written in standard Markdown - include links as appropriate. Screenshots or images representative of what the code does are encouraged, but it's not always possible or applicable, so use discretion.

#### Example of a description

This sample shows you how to use the Azure Cosmos DB for MongoDB API to store and access data from a Flask application.

### Contents

Outline the file contents of the repository (or sample, if it is in its own folder) - this gives a pretty good idea to your customers about what they're dealing with. It also helps customers navigate the codebase, build configuration and any related assets.

The section heading should be **Contents**.

The contents are presented in a table that has two columns - **File/folder** and **Description**. The **File/folder** column contains individual file and folder names, and the **Description** column has a one sentence describing the asset.

The screenshot shows a GitHub repository page for 'sample-template/README.md'. The page includes a table of contents, prerequisites, and setup instructions.

**Contents**

File/folder	Description
src	Sample source code.
.gitignore	Define what to ignore at commit time.
CHANGELOG.md	List of changes to the sample.
CONTRIBUTING.md	Guidelines for contributing to the sample.
README.md	This README file.
LICENSE	The license for the sample.

**Prerequisites**

- Download the [Azure Cosmos DB Emulator](#). The emulator is currently only supported on Windows.
- Install [Visual Studio Code](#) for your platform.
- Install the Don Jayamanne's [Python Extension](#)

**Setup**

- Clone or download this sample repository
- Run the following command to install the required Python modules in the context of the sample folder.

```
pip install -r ./requirements.txt
```

Use absolute links to link individual files and folders to their respective GitHub locations. This ensures that your page renders successfully in the [samples browser](#). It also prevents broken links that result in a 404 error.

#### Example of a repository contents breakdown

File/folder	Description
`src`	Sample source code.
`.gitignore`	Define what to ignore at commit time.
`CHANGELOG.md`	List of changes to the sample.
`CONTRIBUTING.md`	Guidelines for contributing to the sample.
`README.md`	This README file.
`LICENSE`	The license for the sample.

#### Prerequisites

Outline the required components and tools that a customer might need to have on their machine in order to run the sample. This can be anything from frameworks, SDKs, OS versions or IDE releases.

The contents of this section should be presented in a list with relevant links, pointing customers to download the necessary tools or sign up for required services. Don't make the customer figure out on their own where they need to download specific required components and what version is applicable to them - make it abundantly clear in this section.

The section heading should be **Prerequisites**.

The screenshot shows a Microsoft Edge browser window with the URL <https://github.com/dendeli-msft/sample-template/blob/master/README.md>. The page content is as follows:

**Prerequisites**

- Download the [Azure Cosmos DB Emulator](#). The emulator is currently only supported on Windows.
- Install [Visual Studio Code](#) for your platform.
- Install the [Don Jayamanne's Python Extension](#)

**Setup**

1. Clone or download this sample repository
2. Run the following command to install the required Python modules in the context of the sample folder.

```
pip install -r ./requirements.txt
```
3. Open the sample folder in Visual Studio Code or your IDE of choice.

**Running the sample**

1. Make sure the Azure Cosmos DB Emulator is running.
2. Open a terminal window and `cd` to the directory that the app is saved in.
3. Set the environment variable for the Flask app with `set FLASK_APP=app.py` on Windows, or `export FLASK_APP=app.py` if you are using macOS.

## Example of prerequisites

```
- Download the [Azure Cosmos DB Emulator](https://docs.microsoft.com/azure/cosmos-db/local-emulator). The emulator is currently only supported on Windows.
- Install [Visual Studio Code](https://code.visualstudio.com/Download) for your platform.
- Install the Don Jayamanne's [Python Extension](https://marketplace.visualstudio.com/items?itemName=donjayamanne.python)
```

## Setup

Explain how to prepare the sample once the customer clones or downloads the code. The section should outline every step necessary to install dependencies and configure any settings (for example, API keys and output folders).

The section heading should be named **Setup**.

**Prerequisites**

- Download the [Azure Cosmos DB Emulator](#). The emulator is currently only supported on Windows.
- Install [Visual Studio Code](#) for your platform.
- Install the Don Jayamanne's [Python Extension](#)

**Setup**

1. Clone or download this sample repository
2. Run the following command to install the required Python modules in the context of the sample folder.

```
pip install -r .\requirements.txt
```
3. Open the sample folder in Visual Studio Code or your IDE of choice.

**Running the sample**

1. Make sure the Azure Cosmos DB Emulator is running.
2. Open a terminal window and `cd` to the directory that the app is saved in.
3. Set the environment variable for the Flask app with `set FLASK_APP=app.py` on Windows, or `export FLASK_APP=app.py` if you are using macOS.
4. Run the app with `flask run` and point your browser to `http://127.0.0.1:5000/`.
5. Add and remove tasks and see them added and changed in the collection.

**Deploy to Azure**

Do not make assumptions about the fact that the customer already knows what to do or how to install certain packages/dependencies. Assume zero knowledge, and explain as if the person reading the file is starting from scratch.

### Example

```
1. Clone or download this sample repository.
2. Run the following command to install
   the required Python modules in the
   context of the sample folder.
   ```bash
   pip install -r .\requirements.txt
   ```

3. Open the sample folder in Visual Studio Code or your IDE of choice.
```

### Running the sample

Outline step-by-step instructions to execute the sample and see its output. Include steps for executing the sample from the IDE, starting specific services in the Azure portal, or anything related to the overall launch of the code.

The section heading should be **Running the sample**.

sample-template/README.md

GitHub, Inc. (US) | https://github.com/dendeli-msft/sample-template/blob/master/README.md

Onboarding CPS API Docs Team Samples Board Samples Query .NET SDP Board .NET to SDP Query DevRel - Spec Gallery PM@Microsoft

pip install -r ./requirements.txt

3. Open the sample folder in Visual Studio Code or your IDE of choice.

## Running the sample

1. Make sure the Azure Cosmos DB Emulator is running.
2. Open a terminal window and `cd` to the directory that the app is saved in.
3. Set the environment variable for the Flask app with `set FLASK_APP=app.py` on Windows, or `export FLASK_APP=app.py` if you are using macOS.
4. Run the app with `flask run` and point your browser to `http://127.0.0.1:5000/`.
5. Add and remove tasks and see them added and changed in the collection.

## Deploy to Azure

 Deploy to Azure

To deploy this app, you can create a new web app in Azure and enable continuous deployment with a fork of this GitHub repo. Follow the [App Service continuous deployment tutorial](#) to set up continuous deployment with GitHub in Azure.

When deploying to Azure, you should remove your application keys and make sure the section below is not commented out:

```
client = MongoClient(os.getenv("MONGOURL"))
db = client.test      #Select the database
db.authenticate(name=os.getenv("MONGO_USERNAME"),password=os.getenv("MONGO_PASSWORD"))
```

You then need to add your MONGOURL, MONGO\_PASSWORD, and MONGO\_USERNAME to the application settings. You can

As mentioned earlier, try to make no assumptions about the customers' prior knowledge of the platform or target system - explain everything in detail to make our customers successful.

## Example of running instructions

1. Make sure the Azure Cosmos DB Emulator is running.
  2. Open a terminal window and `cd` to the directory that the app is saved in.
  3. Set the environment variable for the Flask app with  
`set FLASK\_APP=app.py` on Windows, or `export FLASK\_APP=app.py`  
if you are using macOS.
  4. Run the app with `flask run` and point your browser to `http://127.0.0.1:5000/`.
  5. Add and remove tasks and see them added and changed in the collection.

## Deploying to Azure

## NOTE

This section is optional, although it is encouraged for code samples that deploy Azure resources.

If it's possible for the sample to be deployed to Azure, it's important that the customer has information on how to do it. Include a button that does a one-click deployment via a pointer to a GitHub repository, as such:

```
<a href="https://deploy.azure.com?repository=https://github.com/heatherbshapiro/To-Do-List---Flask-MongoDB-Example" target="_blank">

</a>
```

The section heading should be **Deploy to Azure**.

The screenshot shows a Microsoft Edge browser window with the URL <https://github.com/dendeli-msft/sample-template/blob/master/README.md>. The page content is as follows:

## Deploy to Azure

To deploy this app, you can create a new web app in Azure and enable continuous deployment with a fork of this GitHub repo. Follow the [App Service continuous deployment tutorial](#) to set up continuous deployment with GitHub in Azure.

When deploying to Azure, you should remove your application keys and make sure the section below is not commented out:

```
client = MongoClient(os.getenv("MONGOURL"))
db = client.test #Select the database
db.authenticate(name=os.getenv("MONGO_USERNAME"),password=os.getenv("MONGO_PASSWORD"))
```

You then need to add your MONGOURL, MONGO\_PASSWORD, and MONGO\_USERNAME to the application settings. You can follow the [website configuration tutorial](#) to learn more about Application Settings in Azure Web Apps.

## Key concepts

Let's take a quick review of what's happening in the app. Open the `app.py` file under the root directory and you find that these lines of code create the Azure Cosmos DB connection. The following code uses the connection string for the local Azure Cosmos DB Emulator. The password needs to be split up as seen below to accommodate for the forward slashes that cannot be parsed otherwise.

- Initialize the MongoDB client, retrieve the database, and authenticate.

```
client = MongoClient("mongodb://127.0.0.1:10250/?ssl=true") #host uri
db = client.test #Select the database
```

This section should include relevant configuration settings, along with potential links to articles where the customer can learn more about tools and services used in deployment.

See an example implementation in the [demo template repository](#).

If you have an [Azure Resource Manager template](#) available, make sure to also set the `azureDeploy` metadata. This will allow us to enable the **Deploy to Azure** capability for the sample in the [samples](#) browser.

## Key concepts

### NOTE

This section is optional, but encouraged. This will give customers understanding about what they can do with the product or service you're targeting with your sample.

Provide additional context on the tools and services used in the sample. Explain some of the code that is being used and how services interact with each other.

The section heading should be **Key concepts**.

You then need to add your MONGOURL, MONGO\_PASSWORD, and MONGO\_USERNAME to the application settings. You can follow the [website configuration tutorial](#) to learn more about Application Settings in Azure Web Apps.

## Key concepts

Let's take a quick review of what's happening in the app. Open the `app.py` file under the root directory and you find that these lines of code create the Azure Cosmos DB connection. The following code uses the connection string for the local Azure Cosmos DB Emulator. The password needs to be split up as seen below to accommodate for the forward slashes that cannot be parsed otherwise.

- Initialize the MongoDB client, retrieve the database, and authenticate.

```
client = MongoClient("mongodb://127.0.0.1:10250/?ssl=true") #host uri
db = client.test      #Select the database
db.authenticate(name="localhost",password='C2y6yDjf5' + r'/R' + '+ob0N8A7Cgv30VRDJWEHLM+4QDU5DE2nQ9nDuVT
```

- Retrieve the collection or create it if it does not already exist.

```
todos = db.todo #Select the collection
```

- Create the app

```
app = Flask(__name__)
title = "TODO with Flask"
heading = "ToDo Reminder"
```

### IMPORTANT

Do not re-write service or product documentation in this section. Where possible and appropriate, point customers to existing official documentation on docs.microsoft.com or a third-party site, if you're working with third-party components.

See an example implementation in the [demo template repository](#).

### Next steps

Outline next steps for the customer to take to learn more about the tools and services being used. Point customers to relevant documentation and other samples, providing an opportunity for them to extend their knowledge of the solutions shown in the sample. This section can be from one sentence to a couple of paragraphs long.

### IMPORTANT

Do not re-write service or product documentation in this section. Where possible and appropriate, point customers to existing official documentation on docs.microsoft.com or a third-party site, if you're working with third-party components.

The section heading should be **Next steps**.

See an example implementation in the [demo template repository](#).

## SEO Considerations

As you are authoring the `README.md`, keep in mind that the contents of the file will make it over to [docs.microsoft.com/samples](#). Familiarize yourself with the [SEO guidelines](#) that apply to your content, to make sure that the sample is easily discoverable through Bing and Google.

# Onboarding Samples to docs.microsoft.com/samples

3/5/2021 • 7 minutes to read

You have a sample that you want customers to discover, so now you want to onboard it to [docs.microsoft.com/samples](#) - congratulations!

Before we start, make sure that you are following the [basic guidelines](#). This ensures that your sample is consistent with the other samples, and will help make sure that customers are successful with your sample.

## Checklist

The following checklist outlines the requirements for samples to be onboarded:

- **YAML front-matter**. Describes the sample and how it will render on [docs.microsoft.com/samples](#).
- **Webhook**. Enables continuous updates and indexing from the repository where the sample code is located.
- **Service account access**. Ensures that we're able to create releases for code samples. Releases are used to give users the option to download the sample without cloning the whole repository.

## Add metadata to readme

For the sample to be indexed in our system, you need to embed [YAML front-matter](#) in your `README.md` file(s).

The front-matter is metadata defining what the sample is, and what information needs to be displayed to end-users on [docs.microsoft.com/samples](#).

There are several conventions for the YAML front-matter that you need to follow:

CONVENTION	DESCRIPTION
One <code>README.md</code> equals one page on <a href="#">docs.microsoft.com/samples</a>	You can have several <code>README.md</code> files in the repository ( <i>for example, one for each sample that you document</i> ) - each will become its own permalink page on the samples portal <i>if</i> indexing is successful for said sample.
Metadata-driven indexing	Only samples that have a <code>README.md</code> with a YAML front-matter containing <code>page_type: sample</code> will be indexed into the system. Any other <code>README.md</code> file that does not have said metadata value will be ignored.

### Metadata structure for YAML front-matter

When we mention YAML front-matter, what we are really referring to is the YAML blurb that gets included in the `README.md` file before all other content in it. We rely on YAML front-matter to guide us through the indexing process and help your customers discover the sample on the [sample browser](#).

As an example, here is how the YAML front-matter would integrate into a Markdown file:

```

---
page_type: sample
languages:
- csharp
- fsharp
- vbnet
products:
- office
- windows
- xbox
---
# My awesome sample on Azure APIs

Hey folks, there is an awesome sample that I want to share...

```

### IMPORTANT

It's key that the YAML front-matter is valid - you can use the GitHub preview capability when issuing a change, or just use a YAML validator tool such as [YAML Lint](#). If the YAML front-matter is invalid, indexing will fail.

### Supported metadata fields for readme.md

Supported metadata fields are outlined below. When you create your front-matter, make sure that you are only using properties that are listed below. If there is any additional metadata that you need to include, insert it as a child node under `extensions` - our indexer is ignoring the contents of that node, while giving you the option to still include information in the sample manifest. Metadata under `extensions` will still be visible in the generated documentation page for use in business intelligence queries and available in the at-rest YAML representation that the page is generated from.

FIELD	REQUIRED?	DESCRIPTION
<code>page_type</code>	Required	Always must be <code>sample</code> . This is used to identify <code>README.md</code> files that need to be ingested.
<code>languages</code>	Required	<p>The array of programming languages covered by the sample. The identifiers need to map to the list of approved programming language values.</p> <p><a href="#">See list of language slugs.</a></p> <p>If the metadata value you need is not on the list, refer to the <a href="#">section on requesting new values</a>.</p>
<code>products</code>	Required	<p>The array of technologies covered by the sample. The identifiers need to map to the list of approved product values.</p> <p><a href="#">See list of product slugs.</a></p> <p>If the metadata value you need is not on the list, refer to the <a href="#">section on requesting new values</a>.</p>

FIELD	REQUIRED?	DESCRIPTION
<code>name</code>	Optional	<p>The name of the sample, as displayed in the list on the <a href="#">samples browser landing page</a>.</p> <p>This is also rendered in the browser title bar for the sample page. Ideally this should be short and to the point. This should be a plain-text value without Markdown or HTML.</p> <p><b>Default value:</b> The first H1 line ( <code>#</code> ). If first H1 line is absent, and the <code>name</code> attribute is missing, indexing of the code sample will fail.</p>
<code>urlFragment</code>	Optional	<p>The URL fragment of the published sample. The full URL will always be <a href="https://docs.microsoft.com/{locale}/samples/{reorganization}/{repo-id}/{urlFragment}">https://docs.microsoft.com/{locale}/samples/{reorganization}/{repo-id}/{urlFragment}</a>. While this metadata node is not required, our recommendation is to <i>always</i> have this value set - this will ensure that the page link is immutable, regardless of sample <code>README.md</code> changes. See below for fragment restrictions.</p> <p><b>Default value:</b> First H1 line ( <code>#</code> ) transformed in a URL-compatible ID. If the first H1 line is absent and the <code>urlFragment</code> attribute is missing, ingestion will fail.</p>
<code>description</code>	Optional	<p>Description of the published sample. Try to keep this simple and no more than 150 characters. This should be a plain-text value without Markdown or HTML.</p> <p><b>Default value:</b> The first 150 characters following the first H1 ( <code>#</code> ) line. If no content is available, no description will be added.</p>
<code>statusNotificationTargets</code>	Optional	[CURRENTLY UNUSED] Array of email aliases to which status updates on the sample build are sent.
<code>azureDeploy</code>	Optional	Fully qualified URL to an Azure Resource Manager template to allow users to deploy the samples directly to Azure. This must be a public URL that is accessible to anyone outside Microsoft.

FIELD	REQUIRED?	DESCRIPTION
<code>extendedZipContent</code>	Deprecated	A list of additional files or folders to include in the ZIP. Each element in <code>extendedZipContent</code> has two properties: <code>path</code> (name in the repo), and <code>target</code> (name in the ZIP). Both are absolute paths. Use <code>/</code> as a directory separator. If your sample uses <code>extendedZipContent</code> , then the Samples service will generate a new public release of your sample on your GitHub project every time the sample is updated.
<code>extensions</code>	Optional	Content that other teams can use to identify samples. The ingestion system will ignore all underlying values. There is no required structure in this node.

#### IMPORTANT

If you do not set `page_type: sample`, the sample will not be indexed in the samples portal.

**Note:** The `urlFragment` may not contain the following characters: `"`, `%`, `<`, `>`, `\`, `/`, `^`, `*`, `:`, `?`, `{`, `}`, `|`, `_`, space, or backquote. Doing so will cause sample ingestion to fail for the entire repository (not just the one with a bad `urlFragment`). For safety, limit yourself to alphanumerics and hyphen.

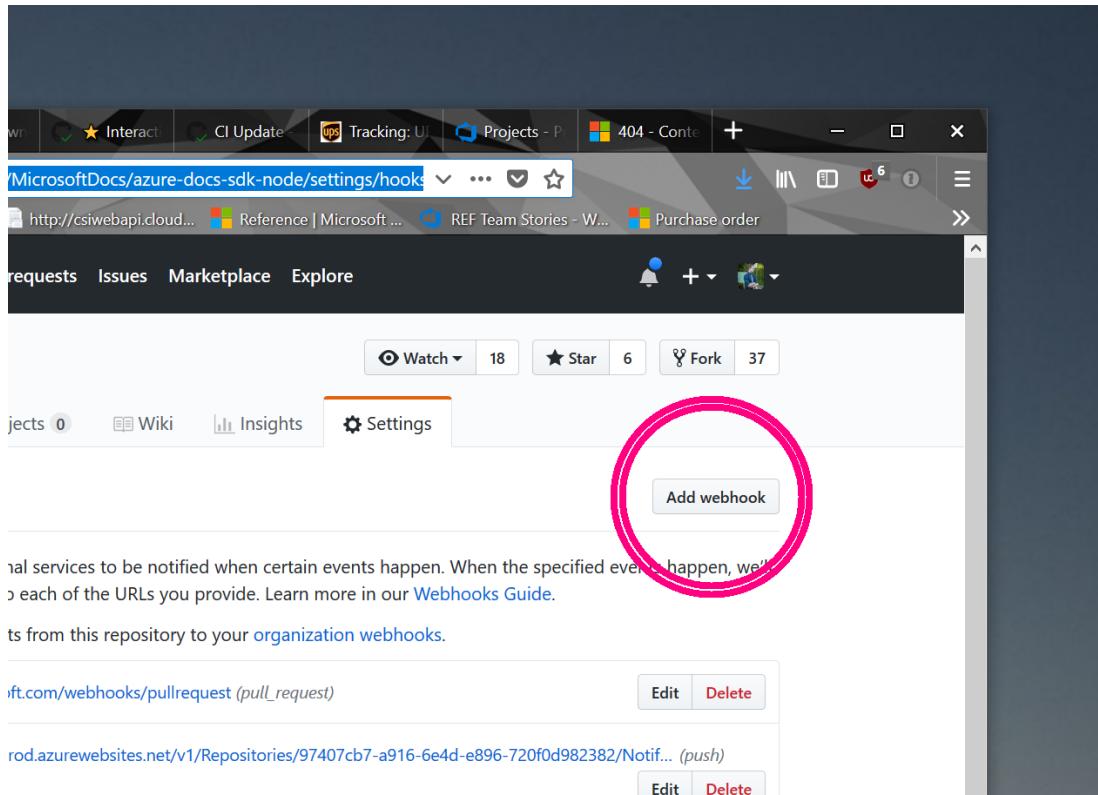
## Create webhook in sample repo

In addition to defining metadata in the `README.md` file, you will need to set up a GitHub webhook in your repository.

1. Open the webhooks page, found on the following URL:

```
https://github.com/{user}/{repo}/settings/hooks
```

2. Select **Add webhook**:



3. Set Content Type to `application/json`.

4. For the URL, use the following:

```
https://samplespayloadcollectionprod.azurewebsites.net/api/CollectPayload
```

5. Specify the following secret:

```
63F187ED-CCD4-4DB5-857C-A4E08E16A520
```

6. For Which events would you like to trigger this webhook? , select the `push` event only.

7. Review hook status using a URL like `https://github.com/{user}/{repo}/settings/hooks/`.

Once the webhook is set up, a test payload will be sent to ensure that the service responds to requests. This test payload is a [zen message](#) and will initially show up as a failed request - do not worry, it's by design and every subsequent request will succeed.

8. Push a new change to the repo to active webhook.

From now on, every change to a sample folder that has a valid `README.md` with proper YAML front-matter will trigger a sample indexing and update, along with the creation of a new release in your sample GitHub repository for each detected code sample.

9. If your webhook fails, check [known issues](#).

## Add account access

Make sure that you add **VSC-Service-Account** with **Write** permissions to the repository. Do not worry about the invite link - the account automatically accepts contributor invites shortly after being added.

### IMPORTANT

Without the service account added, we will not be able to create new sample releases, and the indexing will fail.

## Check sample listing on review site

Once you have everything configured, the code samples marked with YAML metadata are indexed and available on the [on the review site](#). This should take no longer than 10 to 15 minutes. We first publish code on the staged internal site. Due to our automated PR and merging tools, it is then published automatically to production.

If you have any issues with your code samples indexing, [reach out to us directly](#).

### NOTE

[APPLICABLE ONLY TO SRE/DOCS TEAMS] For additional diagnostics guidelines, [refer to the internal document](#).

## How to request new metadata values

It might happen that the product and language values that you want to use are not yet available. There is a straightforward process to request changes. If you need to request an addition, removal, or change to the [product or programming language hierarchy or values](#), please prepare the following information:

1. Primary requestor email.
2. Request summary: (*for example "Add azure-batch to product taxonomy"*)
3. Slug
4. Friendly name
5. Parent entity (*if any*)
6. If you are requesting a change to the structure of the taxonomy (*such as moving an existing term to a different parent*), please provide the following as well:  
a. Business justification for the change: Why now?  
What has changed?  
b. User impact of the change: How will this impact users?  
c. Sponsor: Who supports this change?

Send this in an email to [devrelia@service.microsoft.com](mailto:devrelia@service.microsoft.com) with the subject "*Taxonomy Change Request*". The IA team will batch taxonomy requests on Fridays. Current taxonomy requests are tracked on [aka.ms/metadata-requests](https://aka.ms/metadata-requests).

# Diagnostics for code samples

3/5/2021 • 6 minutes to read

This document is meant to help you better understand how code samples are onboarded to the [samples browser](#), as well as offer guidance on troubleshooting some of the most common issues.

## IMPORTANT

Only samples from the following organizations are supported: [Microsoft](#), [MicrosoftDocs](#), [dotnet](#), [Xamarin](#), [OfficeDev](#), [MicrosoftGraph](#), [Azure](#), [Azure-Samples](#), [aspnet](#), [mspnp](#), [iot-for-all](#), [azure-rtos](#) and [MicrosoftEdge](#). If your organization is not on the list, [contact us](#).

<https://msit.microsoftstream.com/embed/video/d525c670-55b0-4330-8108-a9fdabbe9a8?autoplay=false&showinfo=true>

## Initial onboarding

As described in the [onboarding document](#), the new onboarding process relies on sample owners performing three steps: First, you define metadata. Next, you set up a webhook, followed by giving access to a service account to the samples repository to ensure the content can be indexed.

Once you add the webhook, you might see at first that it failed to perform the call:

The screenshot shows a log entry for a webhook delivery. The URL is <https://samplespayloadcollectionprod.azurewebsites.net/api/CollectPayload>. The status is 400 (Bad Request). The timestamp is 2019-07-17 12:53:55. The payload is a JSON object containing a 'zen' message.

**Request** 9c2bed30-a8cc-11e9-9573-ed18ee202b66

**Response** 400

**Redeliver**

Completed in 3.13 seconds.

**Headers**

```
Request URL: https://samplespayloadcollectionprod.azurewebsites.net/api/CollectPayload
Request method: POST
content-type: application/json
Expect:
User-Agent: GitHub-Hookshot/4104263
X-GitHub-Delivery: 9c2bed30-a8cc-11e9-9573-ed18ee202b66
X-GitHub-Event: ping
X-Hub-Signature: sha1=0a0bfe4b09cf8ec92fcdb104c7acecbbfe1abde8
```

**Payload**

```
{ "zen": "Favor focus over features.", "hook_id": 124742053, "hook": { "type": "Repository", "id": 124742053, "name": "web", "active": true, "events": [ "push" ], "... " } }
```

That behavior is by-design - you are getting a [zen message](#) response on initial provisioning. This is a "heartbeat" message that tells you that the server is responsive to the webhook you set up. You should be able to only get a successful message once you perform a commit in the repository.

#### NOTE

You will not see anything show up in the samples browser after adding the webhook. You need to have at least one `README.md` file committed with the correct metadata, as defined in the [Declaring metadata](#) section of the onboarding guide.

## Adding metadata

Once you add the [required metadata](#) to `README.md` file(s) within the repository, the webhook will fire and will send the information about the changes to the service. At that point, several checks will kick off:

- **Verify that the change originated on the `master` or `main` branch of the repository.** We are currently not loading samples from other branchnames-webhook requests for indexing that are from pushes from other branchnames will be ignored.
- **Verify that the change originated from a public repo.** We do not support private repositories.
- **Verify that `page_type: sample` is present in the YAML metadata.** If this metadata is not included in the `README.md` file, it will be completely ignored by the process.
- **Check whether the YAML metadata is correct to begin with.** If it's malformed, the loading will fail completely.
- **Check that there is a Markdown H1 defined in the sample (name) - it's identified by a single hash (#).** If there is none, the process will look for `name` metadata. If that isn't specified either, the loading of the sample will fail.
- **Check that the values in `languages` are correct and abide to the valid metadata rules.** If the values are not following the expected conventions, the loading process will fail.
- **Check that the values in `products` are correct and abide to the valid metadata rules.** If the values are not following the expected conventions, the loading process will fail.
- **Check that the generated URL fragment for the sample is under 100 characters.** The fragment is generated from the sample name, either defined in the first H1 (#) or from the `name` metadata. Alternatively, the URL fragment can be specified explicitly via the `urlFragment` metadata in the YAML front-matter.
- **Check that there are no duplicate manifest declarations.** If there are (e.g. with duplicate URL fragments), loading will fail.

#### NOTE

We are not checking for the availability of the description in the sample. If you do not specify the `description` metadata, we will take the first 150 characters following the first H1 (#) in the `README.md`. The description in the metadata, shall you specify one, should be under 150 characters as well, otherwise you'll end up with a `null` value in the sample card shown on the [sample browser home page](#).

## Showing up in the samples browser

When your sample has been successfully indexed, we will generate a YAML file representing the content and all relevant metadata in `MicrosoftDocs/samples`. From start to finish, it takes 2-10 minutes for the YAML file to show up, assuming the indexer succeeds.

#### NOTE

If the file does not show up in the repo within 10 minutes after the successful commit in your samples repo, make sure that your repository is set up correctly and check to make sure the YAML front-matter is formatted correctly. If you have still not found the root cause, continue reading this document.

When the YAML is generated, an [OPS build](#) will kick-off to build the content in the `master` branch. Samples will not be live instantly - they will be staged first, until the default branch from the [content repository](#) is merged to `live`. That is a process that happens daily. Until that happens, and dependent on a successful OPS build, your sample should be displayed on the [review \(staging\) site](#).

#### NOTE

It can take up to ~5 minutes after the YAML generation for the sample to show up on the samples browser page.

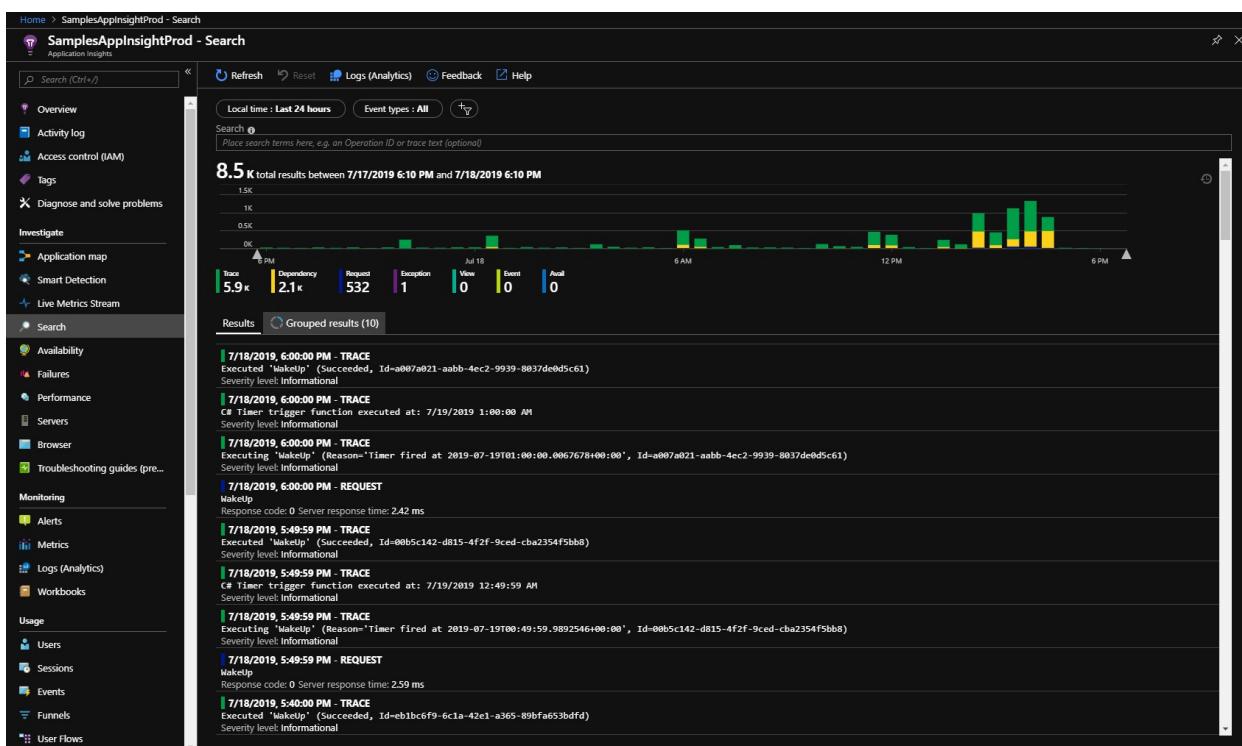
## Checking for detailed diagnostic information

Currently, we do not have a user-friendly way to surface diagnostic information - it's all captured in Application Insights and only available to the Microsoft Samples core team (PM, engineering and SRE). It can be analyzed in [SamplesAppInsightProd - Search](#) instance, in the [DevRel Docs Samples - Production](#) subscription.

#### NOTE

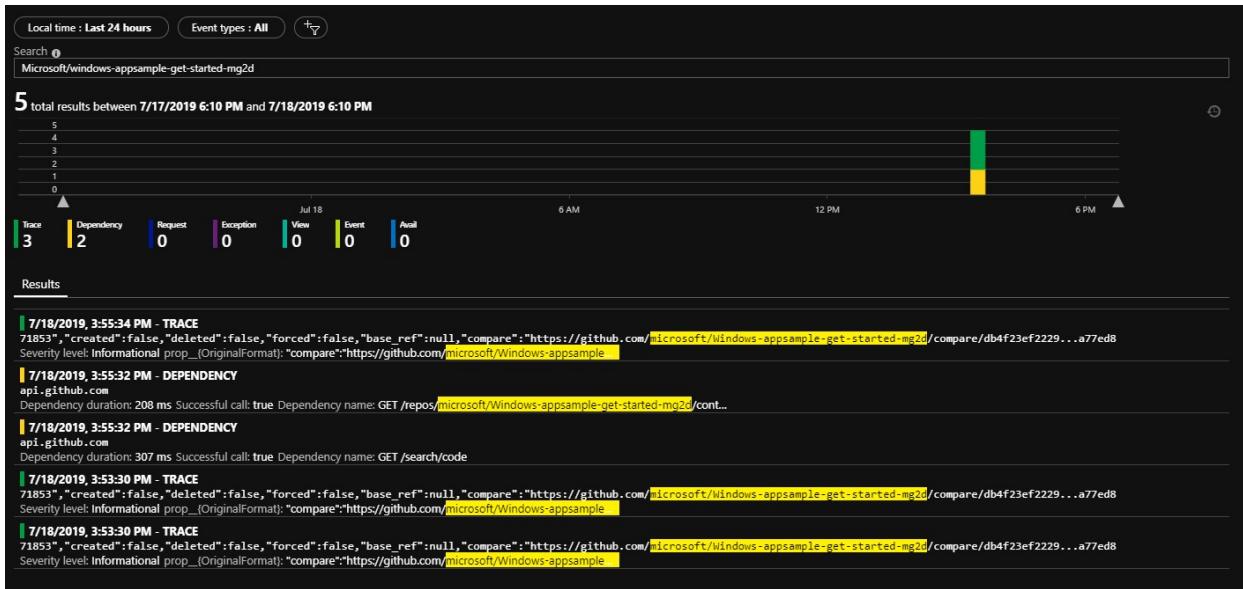
If you are part of the Microsoft Samples core team and do not have access to the resource above, contact [Banani Rath](#).

Make sure that you are in the **Search** view to see all events.



When it comes to figuring out what went wrong, this should be the place to start. The easiest way to get the chain of events is to search for the repository. For example, if you want to understand why the

<https://github.com/Microsoft/windows-appsample-get-started-mg2d> repository fails to onboard, they should search for [Microsoft/windows-appsample-get-started-mg2d](#).



## NOTE

Pay close attention to timestamps for events. You want to be looking for times that closely align with your onboarding changes. Adjust the **Local Time** filter in the Azure Portal if needed.

Once you click on an event, you will see the chain of interconnected events related to the repo onboarding.

If you see an entry with `PAYLOAD_DEAD_LETTER`, that's your first clue that the onboarding failed, because we "dead-letter"-ed the change.

Once you see this, you need to scroll through the list of events, until you find one of type **Error** - that contains the stack trace of what went bad.



Expanding the item and looking at the exception, we can infer the problem - the Application Insights log entry is explicit about the issue:

The screenshot shows the Application Insights search interface. The search results list the error log entry from the previous screenshot. The log entry is highlighted in yellow. The details pane on the right shows the full stack trace of the exception.

Looks like the YAML front-matter is not formatted in the way the indexer expects it. Let's take a look. Opening the `README.md` file in the repository, we notice that the description is not a string but rather an array value:

The screenshot shows the GitHub repository page for `microsoft / Windows-appsample-get-started-mg2`. The `README.md` file is open. The YAML front-matter is highlighted in yellow, showing that the `description` field is an array of strings, which is incorrect for GitHub's indexing requirements.

```

1  ---
2  page_type: sample
3  description:
4    - A sample app accompanying a tutorial that teaches you the basics of game development in MonoGame.
5  languages:
6    - csharp
7  products:
8    - windows
9  statusNotificationTargets:
10   - codefirst@microsoft.com
11  ---
12
13  # Get Started MonoGame 2D
14

```

Once the error is fixed in the `README.md` file, the indexing process will kick-off again, as described earlier.

## Known issues

### Lack of author metadata causes indexing to fail

If you edited a code sample without a GitHub ID (for example, through a local Git client that is set up with an email that is not linked to a GitHub ID), and the sample was previously not edited by anyone with a GitHub account, this will result in the sample being loaded, but the documentation generation build failing.

### Lack of a name causes indexing to fail

In some cases, indexing will fail if the system cannot identify a name for the sample. We use the `name` metadata -OR- the first H1 (`#`) - if neither is found, indexing will fail.

### Lack of secret causes Payload needs signature

Add known secret to webhook and push again.

# Platform-specific guidance

11/2/2020 • 2 minutes to read

As you develop samples for various platforms, it's important to account for best practices for different platforms. The guidance in this section ensures that you have all the tools and knowledge necessary to build the best samples!

## Available guidance

Follow the guides below for considerations around code samples shipped in official [Microsoft-sponsored GitHub organizations](#).

- [Java](#)
- [JavaScript or TypeScript](#)

# Guidance for writing Java samples

11/2/2020 • 2 minutes to read

Whenever you write new Java samples, make sure to follow the guidelines outlined below. While the items below are provided as general recommendations, following them communicates to our community that we understand the Java patterns and practices.

## Java version

[Java 11](#)

## Docker image

[OpenJDK 11 with Alpine](#)

## Framework

[Spring Boot](#)

## Build

[Maven 3+](#)

## General guidelines

Make use of new language features from Java 8 and up to 11 - the practice shows that we understand the language and platform, and aren't providing outdated content. Use code formatting common in Java (*for example, brackets same line, four-space indentation*). For additional information, see [Java Code Conventions](#).

# Guidance for Writing JavaScript Samples

11/2/2020 • 2 minutes to read

This document includes contributions from [John Papa](#), [Brian Clark](#) and [Brian Holt](#).

Whenever you write new JavaScript samples, make sure to follow the guidelines outlined below.

## Process

Use the following `.eslintrc` file:

```
{  
  "extends": ["airbnb-base", "prettier"]  
}
```

Use the following `.prettierrc` file:

```
{  
  "singleQuote": true  
}
```

Make sure to also install the following packages to prepare your environment:

- [eslint](#)
- [eslint-config-airbnb](#)
- [eslint-config-airbnb-base](#)
- [eslint-config-prettier](#)
- [eslint-plugin-import](#)

To make sure that you can leverage the `.prettierrc` configuration, make sure to install the [Prettier Code Formatter](#) for [Visual Studio Code](#).

## Coding style

- Don't use callbacks. Use `async` / `await` when possible, or at least promises.
- Choose between 2 or 4 spaces for tab sizes among the team (can be set in the `.prettierrc` file as `tabWidth` property).
- Standardize on single quotes (set as `singleQuote` property in the `.prettierrc` file).
- Standardize on a `printWidth` value, which determines how many characters will print on a line before wrapping (Prettier defaults to 80, but can choose between that, 100 or 120).

## Node.js version

[Node 8.11.4 \(Active LTS\)](#)

## Docker image

[Node 8.11 Alpine](#)

## Framework

[Express.js](#) for APIs

## Package Tools

npm 5.6.y

## Build

[webpack](#)

## Build scripts

```
npm scripts
```

# Microsoft Docs template finder

4/8/2021 • 2 minutes to read

Documentation should be formatted to best meet a customer's needs. The templates listed below are designed to help you do that.

Find the purpose below that is closest to the reason you're writing your content. Then, use the related template as a starting point for your article.

PURPOSE	TEMPLATE
To explain a service, technology, or functionality from a technical point of view.	<a href="#">Overview</a>
To give new customers an example of some functionality.	<a href="#">Quickstart</a>
To help the customer build a skill by showing them the best way to do something.	<a href="#">Tutorial</a>
To help the customer complete a specific task.	<a href="#">How-to guide</a>
To help the customer understand, rather than do, something.	<a href="#">Concept</a>
To help the customer resolve a specific error or problem that customers commonly run into and would search on.	<a href="#">Troubleshoot</a>

## TIP

These templates are available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

# Template for overview articles

4/16/2021 • 2 minutes to read

This article provides the template in markdown code that you should use when writing an [overview article](#). Use this template when a repository-specific overview template hasn't been defined.

All articles need the [metadata header](#) along with the required values for reporting.

Before you start, make sure an overview is the [correct content type](#) for your article.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown for your overview article:

```
---
```

```
title: #Required; page title is displayed in search results. Include the brand.  
description: #Required; article description that is displayed in search results.  
author: #Required; your GitHub user alias, with correct capitalization.  
ms.author: #Required; microsoft alias of author; optional team alias.  
ms.topic: overview #Required; leave this attribute/value as-is.  
ms.date: #Required; mm/dd/yyyy format.  
ms.custom: template-overview #Required; leave this attribute/value as-is.  
---
```

```
<!--  
Remove all the comments in this template before you sign-off or merge to the  
main branch.  
-->
```

```
<!--  
This template provides the basic structure of a service/product overview article.  
See the [overview guidance](contribute-how-write-overview.md) in the contributor guide.
```

```
To provide feedback on this template contact  
[the templates workgroup](mailto:templateswg@microsoft.com).  
-->
```

```
<!-- 1. H1  
Required. Set expectations for what the content covers, so customers know the  
content meets their needs. H1 format is # What is <product/service>?  
-->
```

```
# What is <product/service>?
```

```
<!-- 2. Introductory paragraph  
Required. Lead with a light intro that describes what the article covers. Answer the  
fundamental “why would I want to know this?” question. Keep it short.  
-->
```

```
[add your introductory paragraph]
```

```
<!-- 3. H2s  
Required. Give each H2 a heading that sets expectations for the content that follows.  
Follow the H2 headings with a sentence about how the section contributes to the whole.  
-->
```

```
## [Section 1 H2]  
<!-- add your content here -->
```

```
## [Section 2 H2]  
<!-- add your content here -->
```

```
## [Section n H2]  
<!-- add your content here -->
```

```
<!-- 4. Next steps  
Required. Provide at least one next step and no more than three. Include some  
context so the customer can determine why they would click the link.  
-->
```

```
## Next steps  
<!-- Add a context sentence for the following links -->  
- [Write an overview](contribute-how-to-write-overview.md)  
- [Links](links-how-to.md)
```

```
<!--  
Remove all the comments in this template before you sign-off or merge to the  
main branch.  
-->
```

# Template for quickstart articles

5/10/2021 • 3 minutes to read

This article provides a quickstart template in markdown code that you should use when writing a [quickstart article](#). Use this template when a repository-specific quickstart template hasn't been defined. For consistency, repository-specific templates should ideally be derived from this global template.

If your quickstart involves an SDK library, refer to [Quickstarts for Azure client libraries](#).

All articles need the [metadata header](#) along with the required values for reporting.

Before you start, make sure a quickstart is the [correct content type](#) for your article.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown to start an article, but remember to delete all the HTML comments before committing to the source repository:

```
---
title: #Required; page title is displayed in search results. Include the brand.
description: #Required; article description that is displayed in search results.
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; microsoft alias of author; optional team alias.
ms.service: #Required; service per approved list. slug assigned by ACOM.
ms.topic: quickstart #Required; leave this attribute/value as-is.
ms.date: #Required; mm/dd/yyyy format.
ms.custom: template-quickstart #Required; leave this attribute/value as-is.
---

<!--
Remove all the comments in this template before you sign-off or merge to the
main branch.
-->

<!--
This template provides the basic structure of a quickstart article.
See the [quickstart guidance](contribute-how-to-mvc-quickstart.md) in the contributor guide.

To provide feedback on this template contact
[the templates workgroup](mailto:templateswg@microsoft.com).
-->

<!-- 1. H1
Required. Starts with "Quickstart: " Make the first word following "Quickstart:" a
verb. Identify both the technology/service and the language or framework, if applicable.
-->

# Quickstart: <do something with X>

<!-- 2. Introductory paragraph
Required. Lead with a light intro that describes what the article covers. Answer the
fundamental "why would I want to know this?" question. Keep it short.
-->


```

[Add your introductory paragraph]

#### <!-- 3. Create a free trial account

Required if a free trial account exists. Include a link to a free trial before the first H2, if one exists. You can find listed examples in [Write quickstart] (contribute-how-to-mvc-quickstart.md)

-->

If you don't have a <service> subscription, create a free trial account...

#### <!-- 4. Prerequisites

Required. First prerequisite is a link to a free trial account if one exists. If there are no prerequisites, state that no prerequisites are needed for this quickstart.

-->

#### ## Prerequisites

- <!-- An Azure account with an active subscription. [Create an account for free] ([https://azure.microsoft.com/free/?WT.mc\\_id=A261C142F](https://azure.microsoft.com/free/?WT.mc_id=A261C142F)). -->
- <!-- prerequisite 2 -->
- <!-- prerequisite n -->

#### <!-- 5. Open Azure Cloud Shell

Optional. Only include the Cloud Shell section if ALL commands can be run in the cloud shell.

-->

#### ## Open Azure Cloud Shell

<!-- [!INCLUDE [cloud-shell-try-it.md](../../../../includes/cloud-shell-try-it.md)] -->

#### <!-- 6. H2s

Required. Prescriptively guide the customer through an end-to-end procedure. Avoid linking off to other content - include whatever the customer needs to complete the scenario in the article.

-->

#### ## [Section 1 heading]

<!-- Introduction paragraph -->

1. Sign in to the [<service> portal](url).

1. <!-- Step 2 -->

1. <!-- Step n -->

#### ## [Section 2 heading]

<!-- Introduction paragraph -->

1. <!-- Step 1 -->

1. <!-- Step 2 -->

1. <!-- Step n -->

#### ## [Section n heading]

<!-- Introduction paragraph -->

1. <!-- Step 1 -->

1. <!-- Step 2 -->

1. <!-- Step n -->

#### <!-- 7. Clean up resources

Required. If resources were created during the quickstart. If no resources were created, state that there are no resources to clean up in this section.

-->

#### ## Clean up resources

If you're not going to continue to use this application, delete <resources> with the following steps:

1. From the left-hand menu...
1. ...click Delete, type...and then click Delete

#### <!-- 8. Next steps

Required: A single link in the blue box format. Point to the next logical quickstart

or tutorial in a series, or, if there are no other quickstarts or tutorials, to some other cool thing the customer can do.

-->

## ## Next steps

Advance to the next article to learn how to create...

> [<div class="nextstepaction"]  
> [Next steps button](contribute-how-to-mvc-quickstart.md)

<!--

Remove all the comments in this template before you sign-off or merge to the main branch.

-->

# Base template for ARM template quickstart articles

5/10/2021 • 5 minutes to read

This article provides a quickstart template in markdown code that you should use when writing an Azure Resource Manager template (ARM template) [quickstart article](#).

All articles need the [metadata header](#) along with the required values for reporting.

You can copy and paste the following Markdown to start an article, but remember to delete all the HTML comments before committing to the source repository:

```
---
title: Create a .... by using Azure Resource Manager template (ARM template)
description: Learn how to create an Azure ... by using Azure Resource Manager template (ARM template).
services: azure-resource-manager
author: your-github-account-name
ms.service: azure-resource-manager
ms.topic: quickstart
ms.custom: subject-armqs
ms.author: your-msft-alias
ms.date: MM/DD/YYYY
---

<!-- ms.topic and ms.custom in the metadata section are required -->

<!--
Remove all the comments from this template before your article is pushed to GitHub and published.
-->

<!-- The H1 must begin with Quickstart: and include the words ARM template. -->

# Quickstart: The H1 heading must include the words ARM template

<!--
First paragraph: Include a sentence that uses Azure Resource Manager template (ARM template) for the first occurrence about the template. For example:
-->

This quickstart describes how to use an Azure Resource Manager template (ARM template) to create \<service>.

<!--
Second paragraph: Use the following include file. This include file is a paragraph that consistently introduces ARM concepts before doing a deployment and includes all our desired links to ARM content. You might need to change the file path of the include file depending on your content structure.
-->

[!INCLUDE [About Azure Resource Manager](../../../../includes/resource-manager-quickstart-introduction.md)] 

<!--
Final paragraph: Explains that readers who are experienced with ARM templates can continue to the deployment. For information about the button image and how to create the template's URI, see "Deploy the template" for Portal.
-->

If your environment meets the prerequisites and you're familiar with using ARM templates, select the **Deploy to Azure** button. The template will open in the Azure portal.

[![Deploy to Azure](../../media/template-deployments/deploy-to-azure.svg)]
(https://portal.azure.com/#create/Microsoft.Template/uri/<template's URI>)
```

```

## Prerequisites

<!--
This section must begin with a sentence that includes a link to create a free Azure account. If your service
has other prerequisites, list them after the free account sentence.
-->

If you don't have an Azure subscription, create a [free account](https://azure.microsoft.com/free/?WT.mc_id=A261C142F) before you begin.

## Review the template

<!--
The first sentence must be the following sentence. Use a link to the quickstart gallery that begins with
https://azure.microsoft.com/resources/templates/.
-->

The template used in this quickstart is from [Azure Quickstart Templates]
(https://azure.microsoft.com/resources/templates/<templateName>).

<!--
After the first sentence, add a JSON code fence that links to the quickstart template. Customers have
provided feedback that they prefer to see the whole template. We recommend you include the entire template
in your article. If your template is too long to show in the quickstart (more than 250 lines), you can
instead add a sentence that says - The template for this article is too long to show here. To view the
template, see [azuredeploy.json](link to template's raw output).

The syntax for the code fence is:
-->

:::code language="json" source="~/quickstart-templates/<TEMPLATE NAME>/azuredeploy.json":::

<!--
After the JSON code fence, a list of each resourceType from the JSON must exist with a link to the template
reference starting with /azure/templates. List the resourceType links in the same order as in the template.

For example:

* [**Microsoft.KeyVault/vaults**](/azure/templates/microsoft.keyvault/vaults): create an Azure key vault.
* [**Microsoft.KeyVault/vaults/secrets**](/azure/templates/microsoft.keyvault/vaults/secrets): create an key
  vault secret.

The URL usually appears as, for example, https://docs.microsoft.com/azure/templates/Microsoft.Network/2019-
11-01/loadBalancers for loadbalancer of Microsoft.Network. Remove the API version from the URL so that the
URL redirects to the latest version.
-->

- [Azure resource type](link to the template reference)
- [Azure resource type](link to the template reference)

<!--
List additional quickstart templates. For example: [Azure Quickstart Templates]
(https://azure.microsoft.com/resources/templates/?resourceType=Microsoft.KeyVault&pageNumber=1&sort=Popular).
Notice the resourceType and sort elements in the URL.
-->

## Deploy the template

<!--
One of the following options must be included:

- **CLI**: In an Azure CLI interactive code fence must contain **az deployment group create**.
- Use Azure CLI version 2.6 or later. To display the version: az --version

For example:

```azurecli-interactive
read -p "Enter a project name that is used for generating resource names:" projectName &&

```

```
read -p "Enter the location (i.e. centralus):" location &&
templateUri="https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-
account-create/azuredeploy.json" &&
resourceGroupName="${{projectName}}rg" &&
az group create --name $resourceGroupName --location "$location" &&
az deployment group create --resource-group $resourceGroupName --template-uri $templateUri &&
echo "Press [ENTER] to continue ..." &&
read
```

```

- **PowerShell**: In an Azure PowerShell interactive code fence must contain **New-AzResourceGroupDeployment**. For example:

```
```azurerepowershell-interactive
$ projectName = Read-Host -Prompt "Enter a project name that is used for generating resource names"
$ location = Read-Host -Prompt "Enter the location (i.e. centralus)"
$ templateUri = "https://raw.githubusercontent.com/Azure/azure-quickstart-templates/master/101-storage-
account-create/azuredeploy.json"
```

```

```
$resourceGroupName = "${{projectName}}rg"

New-AzResourceGroup -Name $resourceGroupName -Location "$location"
New-AzResourceGroupDeployment -ResourceGroupName $resourceGroupName -TemplateUri $templateUri

Read-Host -Prompt "Press [ENTER] to continue ..."
```

```

- **Portal**: Use a button with description **Deploy to Azure**, and the shared image [..../media/template-deployments/deploy-to-azure.svg](#). The template link starts with <https://portal.azure.com/#create/Microsoft.Template/uri/>.

```
```markdown
[![Deploy to Azure](..../media/template-deployments/deploy-to-azure.svg)]
(https://portal.azure.com/#create/Microsoft.Template/uri/https%3A%2F%2Fraw.githubusercontent.com%2FAzure%2Faz-
ure-quickstart-templates%2Fmaster%2F101-key-vault-create%2Fazuredeploy.json)
```

```

To find more information about this deployment option, see [\[Use a deployment button to deploy templates from GitHub repository\]\(https://docs.microsoft.com/azure/azure-resource-manager/templates/deploy-to-azure-button\)](#).

The shared button image is in [\[GitHub\]\(https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/media/template-deployments/deploy-to-azure.svg\)](#).

-->

```
## Review deployed resources
```

<!-- This heading must be titled "Review deployed resources" or "Validate the deployment". -->

<!--

Include at least one method that displays the deployed resources. Use a portal screenshot of the resources, or interactive code fences for Azure CLI (``azurerecli-interactive``) or Azure PowerShell (``azurerepowershell-interactive``).

-->

You can either use the Azure portal to check the deployed resources, or use Azure CLI or Azure PowerShell script to list the deployed resources.

```
## Clean up resources
```

<!--

The Clean up resources section includes a paragraph that explains how to delete unneeded resources. Include at least one method that shows how to clean up resources. Use a portal screenshot, or interactive code fences for Azure CLI (``azurerecli-interactive``) or Azure PowerShell (``azurerepowershell-interactive``).

-->

When no longer needed, delete the resource group, which deletes the resources in the resource group.

<!--

Choose Azure CLI, Azure PowerShell, or Azure portal to delete the resource group.

Here are the samples for Azure CLI and Azure PowerShell:

```
```azurecli-interactive
echo "Enter the Resource Group name:" &&
read resourceGroupName &&
az group delete --name $resourceGroupName &&
echo "Press [ENTER] to continue ..."
```

```azurepowershell-interactive
$resourceGroupName = Read-Host -Prompt "Enter the Resource Group name"
Remove-AzResourceGroup -Name $resourceGroupName
Write-Host "Press [ENTER] to continue..."
```

-->

## Next steps

<!--
Make the next steps similar to other quickstarts and use a blue button to link to the next article for your service. Or direct readers to the article: "Tutorial: Create and deploy your first ARM template" to follow the process of creating a template.

To include additional links for more information about the service, it's acceptable to use a paragraph and bullet points.
-->

For a step-by-step tutorial that guides you through the process of creating a template, see:

> [<a href="/azure/azure-resource-manager/templates/template-tutorial-create-first-template">Tutorial: Create and deploy your first ARM template</a>]
```

# Template for tutorial articles

5/10/2021 • 3 minutes to read

This article provides a tutorial template in markdown code that you should use when [writing a tutorial article](#). Use this template when a repository-specific tutorial template hasn't been defined.

All articles need the [metadata header](#) along with the required values for reporting.

Before you start, make sure a tutorial is the [correct content type](#) for your article.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown for your tutorial article:

```
---
title: #Required; page title is displayed in search results. Include the brand.
description: #Required; article description that is displayed in search results.
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; microsoft alias of author; optional team alias.
ms.service: #Required; service per approved list. slug assigned by ACOM.
ms.topic: tutorial #Required; leave this attribute/value as-is.
ms.date: #Required; mm/dd/yyyy format.
ms.custom: template-tutorial #Required; leave this attribute/value as-is.
---

<!--
Remove all the comments in this template before you sign-off or merge to the
main branch.
-->

<!--
This template provides the basic structure of a tutorial article.
See the [tutorial guidance](contribute-how-to-mvc-tutorial.md) in the contributor guide.

To provide feedback on this template contact
[the templates workgroup](mailto:templateswg@microsoft.com).
-->

<!-- 1. H1
Required. Start with "Tutorial: ". Make the first word following "Tutorial: " a
verb.
-->

# Tutorial: <do something with X>

<!-- 2. Introductory paragraph
Required. Lead with a light intro that describes, in customer-friendly language,
what the customer will learn, or do, or accomplish. Answer the fundamental "why
would I want to do this?" question. Keep it short.
-->

[Add your introductory paragraph]

<!-- 3. Tutorial outline
Required. Use the format provided in the list below.
-->
```

```
-->

In this tutorial, you learn how to:

> [<div class="checklist">]
> * All tutorials include a list summarizing the steps to completion
> * Each of these bullet points align to a key H2
> * Use these green checkboxes in a tutorial

<!-- 4. Prerequisites
Required. First prerequisite is a link to a free trial account if one exists. If there
are no prerequisites, state that no prerequisites are needed for this tutorial.
-->

## Prerequisites

- <!-- An Azure account with an active subscription. [Create an account for free]
  (https://azure.microsoft.com/free/?WT.mc\_id=A261C142F). -->
- <!-- prerequisite 2 -->
- <!-- prerequisite n -->

<!-- 5. H2s
Required. Give each H2 a heading that sets expectations for the content that follows.
Follow the H2 headings with a sentence about how the section contributes to the whole.
-->

## [Section 1 heading]
<!-- Introduction paragraph -->

1. Sign in to the [<service> portal](url).
1. <!-- Step 2 -->
1. <!-- Step n -->

## [Section 2 heading]
<!-- Introduction paragraph -->
1. <!-- Step 1 -->
1. <!-- Step 2 -->
1. <!-- Step n -->

## [Section n heading]
<!-- Introduction paragraph -->
1. <!-- Step 1 -->
1. <!-- Step 2 -->
1. <!-- Step n -->

<!-- 6. Clean up resources
Required. If resources were created during the tutorial. If no resources were created,
state that there are no resources to clean up in this section.
-->

## Clean up resources

If you're not going to continue to use this application, delete
<resources> with the following steps:

1. From the left-hand menu...
1. ...click Delete, type...and then click Delete

<!-- 7. Next steps
Required: A single link in the blue box format. Point to the next logical tutorial
in a series, or, if there are no other tutorials, to some other cool thing the
customer can do.
-->

## Next steps

Advance to the next article to learn how to create...
> [<div class="nextstepaction">]
> [Next steps button](contribute-how-to-mvc-tutorial.md)
```

<!--

Remove all the comments in this template before you sign-off or merge to the  
main branch.

-->

# Template for concept articles

5/10/2021 • 2 minutes to read

This article provides a template in Markdown code that you should use when writing a [concept](#).

All articles need the [metadata header](#) along with the required values for reporting.

Before you start, make sure a concept is the [correct content type](#) for your article.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown to start an article, but remember to delete all the HTML comments before committing to the source repository:

```
---
```

title: #Required; page title is displayed in search results. Include the brand.  
description: #Required; article description that is displayed in search results.  
author: #Required; your GitHub user alias, with correct capitalization.  
ms.author: #Required; microsoft alias of author; optional team alias.  
ms.service: #Required; service per approved list. slug assigned by ACOM.  
ms.topic: conceptual #Required; leave this attribute/value as-is.  
ms.date: #Required; mm/dd/yyyy format.  
ms.custom: template-concept #Required; leave this attribute/value as-is.  
---

<!--Remove all the comments in this template before you sign-off or merge to the main branch.  
-->

<!--  
This template provides the basic structure of a concept article.  
See the [concept guidance](contribute-how-write-concept.md) in the contributor guide.

To provide feedback on this template contact  
[the templates workgroup](mailto:templateswg@microsoft.com).  
-->

<!-- 1. H1  
Required. Set expectations for what the content covers, so customers know the content meets their needs. Should NOT begin with a verb.  
-->

# [H1 heading here]

<!-- 2. Introductory paragraph  
Required. Lead with a light intro that describes what the article covers. Answer the fundamental “why would I want to know this?” question. Keep it short.  
-->

[add your introductory paragraph]

<!-- 3. H2s  
Required. Give each H2 a heading that sets expectations for the content that follows. Follow the H2 headings with a sentence about how the section contributes to the whole.  
-->

## [Section 1 heading]  
<!-- add your content here -->

## [Section 2 heading]  
<!-- add your content here -->

## [Section n heading]  
<!-- add your content here -->

<!-- 4. Next steps  
Required. Provide at least one next step and no more than three. Include some context so the customer can determine why they would click the link.  
-->

## Next steps  
<!-- Add a context sentence for the following links -->  
- [Write concepts](contribute-how-to-write-concept.md)  
- [Links](links-how-to.md)

<!--  
Remove all the comments in this template before you sign-off or merge to the main branch.  
-->

# Template for how-to guides

5/10/2021 • 2 minutes to read

This article provides a template in Markdown code that you should use when writing a [how to guide](#). How-to guides are for *primarily* procedural content that shows a customer how to complete a task in their own environment. How-to guides differ from tutorials in that they can include optional information, explanations, and information to help inform decisions.

All articles need the [metadata header](#) along with the required values for reporting.

Before you start, make sure a how-to guide is the [correct content type](#) for your article.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown to start an article, but remember to delete all the HTML comments before committing to the source repository:

```
---
title: #Required; page title is displayed in search results. Include the brand.
description: #Required; article description that is displayed in search results.
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; microsoft alias of author; optional team alias.
ms.service: #Required; service per approved list. slug assigned by ACOM.
ms.topic: how-to #Required; leave this attribute/value as-is.
ms.date: #Required; mm/dd/yyyy format.
ms.custom: template-how-to #Required; leave this attribute/value as-is.
---

<!--
Remove all the comments in this template before you sign-off or merge to the
main branch.
-->

<!--
This template provides the basic structure of a how-to article.
See the [how-to guidance](contribute-how-to-write-howto.md) in the contributor guide.

To provide feedback on this template contact
[the templates workgroup](mailto:templateswg@microsoft.com).
-->

<!-- 1. H1
Required. Start your H1 with a verb. Pick an H1 that clearly conveys the task the
user will complete.
-->

# [H1 heading here]

<!-- 2. Introductory paragraph
Required. Lead with a light intro that describes, in customer-friendly language,
what the customer will learn, or do, or accomplish. Answer the fundamental "why
would I want to do this?" question. Keep it short.
-->
```

[Add your introductory paragraph]

<!-- 3. Prerequisites

Optional. If you need prerequisites, make them your first H2 in a how-to guide.

Use clear and unambiguous language and use a list format.

-->

## Prerequisites

- <!-- prerequisite 1 -->

- <!-- prerequisite 2 -->

- <!-- prerequisite n -->

<!-- remove this section if prerequisites are not needed -->

<!-- 4. H2s

Required. A how-to article explains how to do a task. The bulk of each H2 should be a procedure.

-->

## [Section 1 heading]

<!-- Introduction paragraph -->

1. <!-- Step 1 -->

1. <!-- Step 2 -->

1. <!-- Step n -->

## [Section 2 heading]

<!-- Introduction paragraph -->

1. <!-- Step 1 -->

1. <!-- Step 2 -->

1. <!-- Step n -->

## [Section n heading]

<!-- Introduction paragraph -->

1. <!-- Step 1 -->

1. <!-- Step 2 -->

1. <!-- Step n -->

<!-- 5. Next steps

Required. Provide at least one next step and no more than three. Include some context so the customer can determine why they would click the link.

-->

## Next steps

<!-- Add a context sentence for the following links -->

- [Write how-to guides](contribute-how-to-write-howto.md)

- [Links](links-how-to.md)

<!--

Remove all the comments in this template before you sign-off or merge to the main branch.

-->

# Change a template or create a new one

5/10/2021 • 3 minutes to read

Your team may have a large collection of content for which the currently available Minimally Viable Content (MVC) templates are relatively unhelpful. You have the following options:

- Request a change to an existing base template. Do this only if you think all teams and products will benefit from it. Create a new bug tracking issue in the Contributor Guide backlog where it will be picked up for review. Detail both the suggested changes and why it should change.
- Create a new base template representing a new article type. Do this only if you think the template will be used globally and frequently by other teams/product areas as well.
- Create a derived template that follows a currently accepted base template as much as it can, differentiating only where absolutely needed.
- Create an entirely new specialized template that fits your team's needs.

For each of these options, the new template requires a review before applying to collections of content. Instructions are detailed below.

## Determine which template type you need

- Base template:

Intended to be used globally across products and teams and doesn't include references specific to any technology, service, or product.[Example base template](#).

- Derived template:

Created when an existing base template is close to what is needed but something different is needed to for a specific product/service/audience. These templates:

- Derive from a base template, but also includes information/design that is specific to a technology area or audience.
- Has metadata that reflects the special requirements and determines how it will appear in the reporting systems.
- Has owners who sign up for notification of a change to the related base "parent" template and resolve differences to keep the derived template up-to-date. If additional changes are made apart from what is inherited from the base template, the template must be re-reviewed and approved.

- Specialized template:

This is a content template so different that it doesn't derive from a standard base template. It is so specific to a product or audience need that it doesn't make sense to suggest as a base template used by all other teams.

- The metadata reflects the special requirements and determines how it will appear in the reporting systems.
- The template needs an owner/author who intends to maintain it.

## Get your new template reviewed and approved

Any Microsoft employee can suggest a new template (base, derived, or specialized) or update an existing one. Start by creating a new User Story tracking item in the Contributor Guide backlog where it will be picked up for review.

- Indicate why it is needed and provide an example.

The review will answer the following questions:

- Would other teams also benefit from this template or something close to it? In that case should it be a standard base template instead to be used globally by other teams.
- Is this a template that might make more sense as a derived template from an existing base template? Or should the existing base template be changed?
- Does the template fit the style and content guidelines? If not, does it need to change, or do the guidelines need to change/improve in some way?
- Does it have the right Metadata?

The final step is review and approval by content team M2's (Angela and Jennie).

## Make your new template available

Once your new template is approved you need to add it to the Contributor Guide repo by creating a pull request (PR). Here are the locations for each template type:

- Base templates: These live in the Contributor Guide repo in the same folder as this article.
- Derived and Specialized templates: These files live in the Contributor Guide repo in a GitHub subfolder that indicates the name of the product or specialized audience.

Here is an example location of a derived tutorial template for SQL.

Content contributors should use the templates in their Product subfolder as a first option. For example, SQL content developers would first check for a SQL specific tutorial template in a SQL subfolder. If one isn't found there, the SQL team uses the base tutorial.

## Template naming conventions

- Base template used globally has file name suffix "--base". Note the two dashes. Example: tutorial-template--base.md
- Derived template used specifically for the SQL product (derived from a base tutorial template) uses "-sql". Note the single dash. Example: tutorial-template-sql.md

## Avoid creating more templates than can be maintained

Templates require maintenance, so it's best to avoid creating too many of them. But Content & Learning does have considerable bodies of work that would benefit from templates that fit their needs. The review process will help make sure that no more templates are created than are needed. And for those that are created, an owner is assigned to keep them up-to-date.

# Azure template for a tutorial article

5/28/2021 • 5 minutes to read

This article provides a Markdown template that you should use when writing an Azure [tutorial article](#).

All articles need the [metadata header](#) along with the required values for reporting. Detailed instructions for all tutorial articles are available in the [Tutorial template article](#).

You can copy and paste the following Markdown for your Azure tutorial article:

```
---
title: Page title has the greatest impact on search
description: The meta-description is not crawled for search rank, but is displayed in the browser.
author: <github account>
ms.author: <MS alias>
ms.service: <service-slug>
ms.topic: tutorial
ms.custom: mvc
ms.date: 11/16/2017

---
<!--Recommended: Remove all the comments in this template before you
sign-off or merge to master.-->

<!--Tutorials are scenario-based procedures for the top customer tasks
identified in milestone one of the
[Content & Learning content model](contribute-get-started-mvc.md).
You only use tutorials to show the single best procedure for completing
an approved top 10 customer task.
-->

# Tutorial: <do something with X>
<!--Required:
Starts with "Tutorial: "
Make the first word following "Tutorial:" a verb.
-->

Introductory paragraph.
<!--Required:
Lead with a light intro that describes, in customer-friendly language,
what the customer will learn, or do, or accomplish. Answer the
fundamental "why would I want to do this?" question.
-->

In this tutorial, you learn how to:

> [<div class="checklist">]
> * All tutorials include a list summarizing the steps to completion
> * Each of these bullet points align to a key H2
> * Use these green checkboxes in a tutorial
<!--Required:
The outline of the tutorial should be included in the beginning and at
the end of every tutorial. These will align to the **procedural** H2
headings for the activity. You do not need to include all H2 headings.
Leave out the prerequisites, clean-up resources and next steps-->

If you don't have a <service> subscription, create a free trial account...
<!-- Required, if a free trial account exists
Because tutorials are intended to help new customers use the product or
service to complete a top task, include a link to a free trial before the
first H2, if one exists. You can find listed examples in
[Write tutorials](contribute-how-to-mvc-tutorial.md)
```

--->

<!--Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.-->

## Prerequisites

- First prerequisite
- Second prerequisite
- Third prerequisite

<!--If you need them, make Prerequisites your first H2 in a tutorial. If there's something a customer needs to take care of before they start (for example, creating a VM) it's OK to link to that content before they begin.-->

## Sign in to <service/product/tool name>

Sign in to the [<service> portal](url).

<!--If you need to sign in to the portal to do the tutorial, this H2 and link are required.-->

## Procedure 1

<!--Required:  
Tutorials are prescriptive and guide the customer through an end-to-end procedure. Make sure to use specific naming for setting up accounts and configuring technology.  
Don't link off to other content - include whatever the customer needs to complete the scenario in the article. For example, if the customer needs to set permissions, include the permissions they need to set, and the specific settings in the tutorial procedure. Don't send the customer to another article to read about it.  
In a break from tradition, do not link to reference topics in the procedural part of the tutorial when using cmdlets or code. Provide customers what they need to know in the tutorial to successfully complete the tutorial.  
For portal-based procedures, minimize bullets and numbering.  
For the CLI or PowerShell based procedures, don't use bullets or numbering.  
-->

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure
  - ![Browser](media/contribute-how-to-mvc-tutorial/browser.png)
  - <!--Use screenshots but be judicious to maintain a reasonable length.  
Make sure screenshots align to the [current standards](https://review.docs.microsoft.com/help/contribute/contribute-how-to-create-screenshot?branch=master).  
If users access your product/service via a web browser the first screenshot should always include the full browser window in Chrome or Safari. This is to show users that the portal is browser-based - OS and browser agnostic.-->
1. Step four of the procedure

## Procedure 2

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure

## Procedure 3

Include a sentence or two to explain only what is needed to complete the procedure.

include a sentence or two to explain only what is needed to complete the procedure.

<!--Code requires specific formatting. Here are a few useful examples of commonly used code blocks. Make sure to use the interactive functionality where possible.

For the CLI or PowerShell based procedures, don't use bullets or numbering.-->

Here is an example of a code block for Java:

```
```java
cluster = Cluster.build(new File("src/remote.yaml")).create();
...
client = cluster.connect();
```

```

or a code block for Azure CLI:

```
```azurecli-interactive
az vm create --resource-group myResourceGroup --name myVM --image win2016datacenter --admin-username
azureuser --admin-password myPassword12
```

```

or a code block for Azure PowerShell:

```
```azurermpowershell-interactive
New-AzureRmContainerGroup -ResourceGroupName myResourceGroup -Name mycontainer -Image
mcr.microsoft.com/windows/servercore/iis:nanoserver -OsType Windows -IpAddressType Public
```

```

## Clean up resources

If you're not going to continue to use this application, delete <resources> with the following steps:

1. From the left-hand menu...
2. ...click Delete, type...and then click Delete

<!--Required:

To avoid any costs associated with following the tutorial procedure, a Clean up resources (H2) should come just before Next steps (H2)

-->

## Next steps

Advance to the next article to learn how to create...

```
> [<div class="nextstepaction">
> [Next steps button](contribute-get-started-mvc.md)
```

<!-- Required:

Tutorials should always have a Next steps H2 that points to the next logical tutorial in a series, or, if there are no other tutorials, to some other cool thing the customer can do. A single link in the blue box format should direct the customer to the next article - and you can shorten the title in the boxes if the original one doesn't fit.

Do not use a "More info section" or a "Resources section" or a "See also section". -->

# Azure template for a manage costs how-to article

4/29/2021 • 11 minutes to read

This article provides a markdown template that you should use when writing an Azure how-to article about [planning to manage costs for an Azure service](#).

The template is designed to give you the freedom needed to write about planning to manage costs for your Azure service. The suggestions in this template are just that: suggestions. Follow the recommendations where possible and disregard any that don't make sense for your situation. Review the comments throughout the template and remove them as you write your article.

You can copy and paste the following markdown for your article:

```
---
title: Plan to manage costs for <ServiceName>
description: Learn how to plan for and manage costs for <AzureServiceName> by using cost analysis in the Azure portal.
author: <GitHubUserName>
ms.author: <MicrosoftAlias>
ms.custom: subject-cost-optimization
ms.service: <serviceslug>
ms.topic: how-to
ms.date: 07/15/2020
---

# Plan to manage costs for <AzureServiceName>

<!-- Check out the following published examples:
- [https://docs.microsoft.com/azure/cosmos-db/plan-manage-costs](https://docs.microsoft.com/azure/cosmos-db/plan-manage-costs)
- [https://docs.microsoft.com/azure/storage/common/storage-plan-manage-costs](https://docs.microsoft.com/azure/storage/common/storage-plan-manage-costs)
- [https://docs.microsoft.com/azure/machine-learning/concept-plan-manage-cost](https://docs.microsoft.com/azure/machine-learning/concept-plan-manage-cost)
-->

<!-- Note for Azure service writer: Links to Cost Management articles are relative links with the ?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn campaign suffix. Please keep the campaign IDs. They're used to measure traffic to Cost Management articles.
-->

<!-- Note for Azure service writer: Modify the following for your service. -->

This article describes how you plan for and manage costs for <AzureServiceName>. First, you use the Azure pricing calculator to help plan for <AzureServiceName> costs before you add any resources for the service to estimate costs. Next, as you add Azure resources, review the estimated costs {if applicable to <AzureServiceName>}. After you've started using <AzureServiceName> resources, use Cost Management features to set budgets and monitor costs. You can also review forecasted costs and identify spending trends to identify areas where you might want to act. Costs for <AzureServiceName> are only a portion of the monthly costs in your Azure bill. Although this article explains how to plan for and manage costs for <ServiceName>, you're billed for all Azure services and resources used in your Azure subscription, including the third-party services.

## Prerequisites

<!--Note for Azure service writer: The section covers prereqs for the cost analysis feature. Add other prereqs needed for your service. -->

Cost analysis in Cost Management supports most Azure account types, but not all of them. To view the full
```

list of supported account types, see [Understand Cost Management data](../cost-management-billing/costs/understand-cost-mgt-data.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn). To view cost data, you need at least read access for an Azure account. For information about assigning access to Azure Cost Management data, see [Assign access to data](../cost-management/assign-access-acm-data.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn).

<!--Note for Azure service writer: If you have other prerequisites for your service, insert them here -->

<!--Note for Azure service writer: Modify the following H2 sections for your service. -->

## Estimate costs before using <AzureServiceName>

- Use the [Azure pricing calculator](https://azure.microsoft.com/pricing/calculator/) to estimate costs before you add <AzureServiceName>.

<!-- Note for Azure service writer: At a minimum, insert a brief walkthrough of using the calculator for your service. You don't need more than a couple of paragraphs. Add screenshots where useful. Add a screenshot where the estimated cost is shown. -->

<!--Note to Azure service writer: Replace the following example image with one specific to your service. -->

:::image type="content" source="../../media/contribute-how-to-write-cost-management-conceptual-article/pricing-calc.png" alt-text="Example showing estimated cost in the Azure Pricing calculator."lightbox="../../media/contribute-how-to-write-cost-management-conceptual-article/pricing-calc.png" :::

## Understand the full billing model for <AzureServiceName>

<AzureServiceName> runs on Azure infrastructure that accrues costs when you deploy new resources. It's important to understand that there could be other additional infrastructure costs that might accrue.

<!--Note to Azure service writer: Include each of the following subsections at a minimum -->

### How you're charged for <AzureServiceName>

<!--Note to Azure service writer: Tell users how they're charged for your service. We don't need great detail, at a minimum:

Tell users about the billable meters and units of measure for your service.

Billable meters are the individual components of your service that appear on the customer's bill and are also shown in cost analysis under your service. We need to give a brief explanation about the meters used by your service. At a minimum, list the meters for your service and talk about the unit of measure for each one.

A unit of measure varies greatly among Azure services. It could be:

- Time-based like seconds, minutes, hours, and so on
- Size based, KB, MB, GB, and so on
- Number of transactions

-->

When you create or use <AzureServiceName> resources, you might get charged for the following meters:

- <Meter01> - You're charged for it based on the number of <UnitOfMeasure>.
- <Meter02> - You're charged for it based on the number of <UnitOfMeasure>.
- And so on

At the end of your billing cycle, the charges for each meter are summed. Your bill or invoice shows a section for all <AzureServiceName> costs. There's a separate line item for each meter.

### Other costs that might accrue with <AzureServiceName>

<!--Note to Azure service writer: Include any costs that aren't obvious, hidden, or otherwise might not be present in the pricing calculator or resource creation experience in the Azure portal. You might need to sync with your product team to identify hidden costs. If you're certain that costs accrue only for your service and no others, then omit this section. -->

When you create resources for <AzureServiceName>, resources for other Azure services are also created. They

When you create resources for <AzureServiceName>, resources for other Azure services are also created. They include:

- <OtherAzureService1>
- <OtherAzureService2>

### Costs might accrue after resource deletion

<!--Note to Azure service writer: You might need to sync with your product team to identify resources that continue to exist after those ones for your service are deleted. If you're certain that no resources can exist after those for your service are deleted, then omit this section. -->

After you delete <AzureServiceName> resources, the following resources might continue to exist. They continue to accrue costs until you delete them.

- <OtherServiceResource1>
- <OtherServiceResource2>

### Using Azure Prepayment with <AzureServiceName>

<!--Note to Azure service writer: Let the user know that most 1st party Azure service charges can be fulfilled by Azure Prepayment (previously called EA monetary commitment credit). However, charges from third party products and services including those from the Azure Marketplace cannot be paid for by Azure Prepayment credit. -->

You can pay for <AzureServiceName> charges with your Azure Prepayment credit. However, you can't use Azure Prepayment credit to pay for charges for third party products and services including those from the Azure Marketplace.

## Review estimated costs in the Azure portal

<!-- Note for Azure service writer: If your service shows estimated costs when a user is creating resources in the Azure portal, at a minimum, insert this section as a brief walkthrough that steps through creating a <AzureServiceName> resource where the estimated cost is shown to the user, updated for your service. Add a screenshot where the estimated costs or subscription credits are shown.

If your service doesn't show costs as they create a resource or if estimated costs aren't shown to users before they use your service, then omit this section.

For example, you might start with the following (modify for your service):

-->

As you create resources for <AzureServiceName>, you see estimated costs.

To create a <ResourceName> and view the estimated price:

1. Navigate to the service in the Azure portal.
2. Create the resource.
3. Review the estimated price shown in the summary.
4. Finish creating the resource.

<!-- Note to Azure service writer: Replace the following example image with one specific to your service. Ensure that you do not show UNIT pricing. Total pricing is okay to show. If you show total pricing, don't show the number of units. -->

:::image type="content" source="../../../../media/contribute-how-to-write-cost-management-conceptual-article/create-resource.png" alt-text="Example showing estimated costs while creating a resource."lightbox="../../../../media/contribute-how-to-write-cost-management-conceptual-article/create-resource.png" :::

<!--Note to Azure service writer: Add a paragraph like: -->

If your Azure subscription has a spending limit, Azure prevents you from spending over your credit amount. As you create and use Azure resources, your credits are used. When you reach your credit limit, the resources that you deployed are disabled for the rest of that billing period. You can't change your credit limit, but you can remove it. For more information about spending limits, see [Azure spending limit] (./cost-management-billing/manage/spending-limit.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_inproduct-learn).

## Monitor costs

```
<!-- Note to Azure service writer: Modify the following as needed for your service. Replace example screenshots with ones taken for your service. If you need assistance capturing screenshots, ask banders for help. -->
```

As you use Azure resources with <ServiceName>, you incur costs. Azure resource usage unit costs vary by time intervals (seconds, minutes, hours, and days) or by unit usage (bytes, megabytes, and so on.) As soon as <ServiceName> use starts, costs are incurred and you can see the costs in [cost analysis](../cost-management/quick-acm-cost-analysis.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn).

When you use cost analysis, you view <ServiceName> costs in graphs and tables for different time intervals. Some examples are by day, current and prior month, and year. You also view costs against budgets and forecasted costs. Switching to longer views over time can help you identify spending trends. And you see where overspending might have occurred. If you've created budgets, you can also easily see where they're exceeded.

To view <ServiceName> costs in cost analysis:

1. Sign in to the Azure portal.
2. Open the scope in the Azure portal and select \*\*Cost analysis\*\* in the menu. For example, go to \*\*Subscriptions\*\*, select a subscription from the list, and then select \*\*Cost analysis\*\* in the menu. Select \*\*Scope\*\* to switch to a different scope in cost analysis.
3. By default, cost for services are shown in the first donut chart. Select the area in the chart labeled <ServiceName>.

Actual monthly costs are shown when you initially open cost analysis. Here's an example showing all monthly usage costs.

```
:::image type="content" source="../../media/contribute-how-to-write-cost-management-conceptual-article/all-costs.png" alt-text="Example showing accumulated costs for a subscription."  
lightbox="../../media/contribute-how-to-write-cost-management-conceptual-article/all-costs.png" :::
```

```
<!-- Note to Azure service writer: This example shows costs for an example Azure subscription. You can see service costs for App Service, Storage, Backup, Virtual Networks, and Advanced Threat Protection. Replace this example image with one that shows costs for your service. Your screenshot should look like the one above. -->
```

To narrow costs for a single service, like <ServiceName>, select \*\*Add filter\*\* and then select \*\*Service name\*\*. Then, select \*\*<ServiceName>\*\*.

Here's an example showing costs for just <ServiceName>.

```
:::image type="content" source="../../media/contribute-how-to-write-cost-management-conceptual-article/service-specific-cost.png" alt-text="Example showing accumulated costs for ServiceName."  
lightbox="../../media/contribute-how-to-write-cost-management-conceptual-article/service-specific-cost.png" :::
```

```
<!-- Note to Azure service writer: The image shows an example for Azure Storage. Replace the example image with one that shows costs for your service. -->
```

In the preceding example, you see the current cost for the service. Costs by Azure regions (locations) and <ServiceName> costs by resource group are also shown. From here, you can explore costs on your own.

## Create budgets

```
<!-- Note to Azure service writer: Modify the following as needed for your service. -->
```

You can create [budgets](../cost-management/tutorial-acm-create-budgets.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn) to manage costs and create [alerts](../cost-management-billing/costs/cost-mgt-alerts-monitor-usage-spending.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn) that automatically notify stakeholders of spending anomalies and overspending risks. Alerts are based on spending compared to budget and cost thresholds. Budgets and alerts are created for Azure subscriptions and resource groups, so they're useful as part of an overall cost monitoring strategy.

Budgets can be created with filters for specific resources or services in Azure if you want more granularity present in your monitoring. Filters help ensure that you don't accidentally create new resources that cost you additional money. For more information about the filter options available when you create a budget, see [Group and filter options](../cost-management-billing/costs/group-filter.md?WT.mc\_id=costmanagementcontent\_docsacmhorizontal\_-inproduct-learn).

```
WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn).

## Export cost data

You can also [export your cost data](../cost-management-billing/costs/tutorial-export-acm-data.md?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn) to a storage account. This is helpful when you need or others to do additional data analysis for costs. For example, a finance teams can analyze the data using Excel or Power BI. You can export your costs on a daily, weekly, or monthly schedule and set a custom date range. Exporting cost data is the recommended way to retrieve cost datasets.

## Other ways to manage and reduce costs for <ServiceName>

<!-- Note to Azure service writer: This is an optional section. Other than using the Cost Management methods above, there are probably ways to minimize costs for your service that are specific to your service. Because customers only pay for what they use and when they use less of a resource, the result is a smaller bill. You might already have published cost-saving content. For example, you might have best practice advice or specific ways to reduce costs that are specific to your service. If so, try to add that guidance here or at least summarize key points. Try to be as prescriptive as possible. If you have more comprehensive content, add links to your other published articles or sections here.

Add a statement that discusses any recommended settings for your service that might help keep the charges minimal if a service isn't being actively used by the customer. For example: Will turning off a VM help to get no charges for the specific VM resource?

If your team has no cost-saving recommendations or best practice advice to reduce costs, then cut this section.

-->

## Next steps

- Learn [how to optimize your cloud investment with Azure Cost Management](../cost-management-billing/cost-mgt-best-practices.md?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn).
- Learn more about managing costs with [cost analysis](../cost-management-billing/costs/quick-acm-cost-analysis.md?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn).
- Learn about how to [prevent unexpected costs](../cost-management-billing/understand/analyze-unexpected-charges.md?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn).
- Take the [Cost Management](/learn/parts/control-spending-manage-bills?WT.mc_id=costmanagementcontent_docsacmhorizontal_-inproduct-learn) guided learning course.

<!-- Insert links to other articles that might help users save and manage costs for your service here.

Create a table of contents entry for the article in the How-to guides section where appropriate. -->
```

# Base templates for Azure service monitoring article

11/2/2020 • 18 minutes to read

This article provides two monitoring templates meant to act as a starting place for anyone wanting to know how to monitor an Azure service that has onboarded to Azure Monitor. Even if your Azure service does not use Azure Monitor, it's best that you have articles named like those below so we are consistent across Azure, even if you don't use the same template format.

If your service is named "Azure Service Fabric", your article names should be

- Monitor Azure Service Fabric
- Monitor Azure Service Fabric data reference

For more information, see [writing monitoring guidance for your Azure Monitor enabled-service](#) which details why we are doing this work.

All articles need the [metadata header](#) along with the required values for reporting.

You can copy and paste the following Markdown as the starting point for your *Monitor <servicename>* and *Monitor <servicename> data reference* articles. The templates themselves will not validate correctly as-is. This is on-purpose to make sure that you fill in various details as relevant to your service.

## Monitor [servicename] article

```
---
title: Monitoring [TODO-service-name] #Required; Must be "Monitoring *your official service name*
description: Start here to learn how to monitor [TODO-service-name] #Required;
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; Microsoft alias of author; optional team alias.
ms.service: #Required; The service you are monitoring
ms.custom: subject-monitoring
ms.date: #Required; mm/dd/yyyy format.
---

<!-- VERSION 2.2
Template for the main monitoring article for Azure services.
Keep the required sections and add/modify any content for any information specific to your service.
This article should be in your TOC with the name *monitor-[TODO-replace-with-service-name].md* and the TOC title "Monitor [TODO-replace-with-service-name]".
Put accompanying reference information into an article in the Reference section of your TOC with the name
*monitor-[service-name]-reference.md* and the TOC title "Monitoring data".
Keep the headings in this order.
-->

<!-- IMPORTANT STEP 1. Do a search and replace of [TODO-replace-with-service-name] with the name of your
service. That will make the template easier to read -->

# Monitoring [TODO-replace-with-service-name]
<!-- REQUIRED. Please keep headings in this order -->
<!-- Most services can use this section unchanged. Add to it if there are any unique charges if your service
has significant monitoring beyond Azure Monitor. -->

When you have critical applications and business processes relying on Azure resources, you want to monitor
those resources for their availability, performance, and operation.

This article describes the monitoring data generated by [TODO-replace-with-service-name]. [TODO-replace-
with-service-name] uses [Azure Monitor](/azure/azure-monitor/overview). If you are unfamiliar with the
features of Azure Monitor common to all Azure services that use it, read [Monitoring Azure resources with
Azure Monitor](/azure/azure-monitor/essentials/monitor-azure-resources)
```

Azure Monitor](/azure/azure-monitor/essentials/monitor-azure-resource).

<!-- Optional diagram showing monitoring for your service. If you need help creating one, contact robb@microsoft.com -->

## Monitoring overview page in Azure portal  
<!-- OPTIONAL. Please keep headings in this order -->  
<!-- If you don't have an over page, remove this section. If you keep it, edit it if there are any unique charges if your service has significant monitoring beyond Azure Monitor. -->

The \*\*Overview\*\* page in the Azure portal for each \*[Service resource]\* includes \*[provide a description of the data in the Overview page.]\*.

## \*[TODO-replace-with-service-name]\* insights

<!-- OPTIONAL SECTION. Only include if your service has an "insight" associated with it. Examples of insights include  
- CosmosDB <https://docs.microsoft.com/azure/azure-monitor/insights/cosmosdb-insights-overview>  
- If you still aren't sure, contact azmondocs@microsoft.com.>  
-->

Some services in Azure have a special focused pre-built monitoring dashboard in the Azure portal that provides a starting point for monitoring your service. These special dashboards are called "insights".

<!-- Give a quick outline of what your "insight page" provides and refer to another article that gives details -->

## Monitoring data

<!-- REQUIRED. Please keep headings in this order -->  
[TODO-replace-with-service-name] collects the same kinds of monitoring data as other Azure resources that are described in [Monitoring data from Azure resources](/azure/azure-monitor/insights/monitor-azure-resource#monitoring-data-from-Azure-resources).

See [Monitoring \*[TODO-replace-with-service-name]\* data reference](monitor-service-reference.md) for detailed information on the metrics and logs metrics created by [TODO-replace-with-service-name].

<!-- If your service has additional non-Azure Monitor monitoring data then outline and refer to that here. Also include that information in the data reference as appropriate. -->

## Collection and routing

<!-- REQUIRED. Please keep headings in this order -->

Platform metrics and the Activity log are collected and stored automatically, but can be routed to other locations by using a diagnostic setting.

Resource Logs are not collected and stored until you create a diagnostic setting and route them to one or more locations.

<!-- Include any additional information on collecting logs. The number of things that diagnostics settings control is expanding -->

See [Create diagnostic setting to collect platform logs and metrics in Azure](/azure/azure-monitor/platform/diagnostic-settings) for the detailed process for creating a diagnostic setting using the Azure portal, CLI, or PowerShell. When you create a diagnostic setting, you specify which categories of logs to collect. The categories for \*[TODO-replace-with-service-name]\* are listed in [[TODO-replace-with-service-name] monitoring data reference](monitor-service-reference.md#resource-logs).

<!-- OPTIONAL: Add specific examples of configuration for this service. For example, CLI and PowerShell commands for creating diagnostic setting. Ideally, customers should set up a policy to automatically turn on collection for services. Azure monitor has Resource Manager template examples you can point to. See <https://docs.microsoft.com/azure/azure-monitor/samples/resource-manager-diagnostic-settings>. Contact azmondocs@microsoft.com if you have questions. -->

The metrics and logs you can collect are discussed in the following sections.

```
## Analyzing metrics
```

```
<!-- REQUIRED. Please keep headings in this order  
If you don't support metrics, say so. Some services may be only onboarded to logs -->
```

```
You can analyze metrics for *[TODO-replace-with-service-name]* with metrics from other Azure services using metrics explorer by opening **Metrics** from the **Azure Monitor** menu. See [Getting started with Azure Metrics Explorer](/azure/azure-monitor/platform/metrics-getting-started) for details on using this tool.
```

```
<!-- Point to the list of metrics available in your monitor-service-reference article. -->  
For a list of the platform metrics collected for [TODO-replace-with-service-name], see [Monitoring * [service-name]* data reference metrics](monitor-service-reference.md#metrics)
```

```
<!-- REQUIRED for services that use a Guest OS. That includes agent based services like Virtual Machines, Service Fabric, Cloud Services, and perhaps others. Delete the section otherwise -->  
Guest OS metrics must be collected by agents running on the virtual machines hosting your service. <!-- Add additional information as appropriate -->. For more information, see [Overview of Azure Monitor agents](/azure/azure-monitor/platform/agents-overview)
```

```
For reference, you can see a list of [all resource metrics supported in Azure Monitor](/azure/azure-monitor/platform/metrics-supported).
```

```
<!-- Optional: Call out additional information to help your customers. For example, you can include additional information here about how to use metrics explorer specifically for your service. Remember that the UI is subject to change quite often so you will need to maintain these screenshots yourself if you add them in. -->
```

```
## Analyzing logs
```

```
<!-- REQUIRED. Please keep headings in this order  
If you don't support resource logs, say so. Some services may be only onboarded to metrics and the activity log. -->
```

```
Data in Azure Monitor Logs is stored in tables where each table has its own set of unique properties.
```

```
All resource logs in Azure Monitor have the same fields followed by service-specific fields. The common schema is outlined in [Azure Monitor resource log schema](https://docs.microsoft.com/azure/azure-monitor/platform/diagnostic-logs-schema#top-level-resource-logs-schema) The schema for [service name] resource logs is found in the [[TODO-replace-with-service-name] Data Reference](monitor-service-reference.md#schemas)
```

```
The [Activity log](/azure/azure-monitor/platform/activity-log) is a type of platform log in Azure that provides insight into subscription-level events. You can view it independently or route it to Azure Monitor Logs, where you can do much more complex queries using Log Analytics.
```

```
For a list of the types of resource logs collected for [TODO-replace-with-service-name], see [Monitoring [TODO-replace-with-service-name] data reference](monitor-service-reference.md#resource-logs)
```

```
For a list of the tables used by Azure Monitor Logs and queryable by Log Analytics, see [Monitoring [TODO-replace-with-service-name] data reference](monitor-service-reference.md##azure-monitor-logs-tables)
```

```
<!-- Optional: Call out additional information to help your customers. For example, you can include additional information here about log usage or what logs are most important. Remember that the UI is subject to change quite often so you will need to maintain these screenshots yourself if you add them in. -->
```

```
### Sample Kusto queries
```

```
<!-- REQUIRED if you support logs. Please keep headings in this order -->  
<!-- Add sample Log Analytics Kusto queries for your service. -->
```

```
> [<strong>!IMPORTANT</strong>]
```

```
> When you select **Logs** from the [service-name] menu, Log Analytics is opened with the query scope set to the current [Service resource]. This means that log queries will only include data from that resource. If you want to run a query that includes data from other [resource] or data from other Azure services, select **Logs** from the **Azure Monitor** menu. See [Log query scope and time range in Azure Monitor Log Analytics](/azure/azure-monitor/log-query/scope/) for details.
```

```
<!-- REQUIRED: Include queries that are helpful for figuring out the health and state of your service. Ideally, use some of these queries in the alerts section. It's possible that some of your queries may be in
```

the Log Analytics UI (sample or example queries). Check if so. -->

Following are queries that you can use to help you monitor your [Service] resource.

```
<!-- Put in a code section here. -->
```Kusto
```
## Alerts

<!-- SUGGESTED: Include useful alerts on metrics, logs, log conditions or activity log. Ask your PMs if you don't know.
This information is the BIGGEST request we get in Azure Monitor so do not avoid it long term. People don't know what to monitor for best results. Be prescriptive
-->

Azure Monitor alerts proactively notify you when important conditions are found in your monitoring data. They allow you to identify and address issues in your system before your customers notice them. You can set alerts on [metrics](/azure/azure-monitor/platform/alerts-metric-overview), [logs](/azure/azure-monitor/platform/alerts-unified-log), and the [activity log](/azure/azure-monitor/platform/activity-log-alerts). Different types of alerts have benefits and drawbacks

<!-- only include next line if applications run on your service and work with App Insights. --> If you are creating or running an application which run on <*service*> [Azure Monitor Application Insights](/azure/azure-monitor/overview#application-insights) may offer additional types of alerts.
<!-- end -->
```

The following table lists common and recommended alert rules for [service-name].

```
<!-- Fill in the table with metric and log alerts that would be valuable for your service. Change the format as necessary to make it more readable -->
Alert type	Condition	Description
```

## Next steps

```
<!-- Add additional links. You can change the wording of these and add more if useful. -->
```

- See [Monitoring [service-name] data reference](monitor-service-reference.md) for a reference of the metrics, logs, and other important values created by [service name].  
\*>.
- See [Monitoring Azure resources with Azure Monitor](/azure/azure-monitor/insights/monitor-azure-resource) for details on monitoring Azure resources.

## Monitor [servicename] data reference article

```
---
title: Monitoring [TODO-replace-with-service-name] data reference #Required; *your official service name*
description: Important reference material needed when you monitor [TODO-replace-with-service-name]
author: #Required; your GitHub user alias, with correct capitalization.
ms.topic: reference
ms.author: #Required; Microsoft alias of author; optional team alias.
ms.service: #Required; service you are monitoring
ms.custom: subject-monitoring
ms.date: #Required; mm/dd/yyyy format.
---
<!-- VERSION 2.3
Template for monitoring data reference article for Azure services. This article is support for the main "Monitoring [servicename]" article for the service. -->

<!-- IMPORTANT STEP 1. Do a search and replace of [TODO-replace-with-service-name] with the name of your service. That will make the template easier to read -->
```

```

# Monitoring [TODO-replace-with-service-name] data reference

See [Monitoring [TODO-replace-with-service-name]](monitor-service.md) for details on collecting and
analyzing monitoring data for [TODO-replace-with-service-name].
```

## Metrics

<!-- REQUIRED if you support Metrics. If you don't, keep the section but call that out. Some services are only onboarded to logs.  
<!-- Please keep headings in this order -->

<!-- 2 options here depending on the level of extra content you have. -->

-----\*\*OPTION 1 EXAMPLE\*\* -----

<!-- OPTION 1 - Minimum - Link to relevant bookmarks in https://docs.microsoft.com/azure/azure-monitor/platform/metrics-supported, which is auto generated from underlying systems. Not all metrics are published depending on whether your product group wants them to be. If the metric is published, but descriptions are wrong or missing, contact your PM and tell them to update them in the Azure Monitor "shoebox" manifest. If this article is missing metrics that you and the PM know are available, both of you contact azmondocs@microsoft.com.  
-->

<!-- Example format. There should be AT LEAST one Resource Provider/Resource Type here. -->

This section lists all the automatically collected platform metrics collected for [TODO-replace-with-service-name].

|                           |                                                                                                                                    |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| Metric Type               | Resource Provider / Type Namespace<br/> and link to individual metrics                                                             |
| ----- -----               |                                                                                                                                    |
| Virtual Machine           | [Microsoft.Compute/virtualMachine](/azure/azure-monitor/platform/metrics-supported#microsoftcomputevirtualmachines)                |
| Virtual machine scale set | [Microsoft.Compute/virtualMachinescaleset](/azure/azure-monitor/platform/metrics-supported#microsoftcomputevirtualmachinescaleset) |

-----\*\*OPTION 2 EXAMPLE\*\* -----

<!-- OPTION 2 - Link to the metrics as above, but work in extra information not found in the automated metric-supported reference article. NOTE: YOU WILL NOW HAVE TO MANUALLY MAINTAIN THIS SECTION to make sure it stays in sync with the metrics-supported link. For highly customized example, see [CosmosDB](https://docs.microsoft.com/azure/cosmos-db/monitor-cosmos-db-reference#metrics). They even regroup the metrics into usage type vs. resource provider and type.  
-->

<!-- Example format. Mimic the setup of metrics supported, but add extra information -->

### Virtual Machine metrics

Resource Provider and Type: [Microsoft.Compute/virtualMachines](/azure/azure-monitor/platform/metrics-supported#microsoftcomputevirtualmachines)

|                         |      |                                                                    |                                                  |  |
|-------------------------|------|--------------------------------------------------------------------|--------------------------------------------------|--|
| Metric                  | Unit | Description                                                        | *TODO replace this label with other information* |  |
| ----- ----- ----- ----- |      |                                                                    |                                                  |  |
|                         |      | Use this metric for <!-- put your specific information in here --> |                                                  |  |
|                         |      |                                                                    |                                                  |  |

### Virtual machine scale set metrics

Namespace- [Microsoft.Compute/virtualMachinescaleset](/azure/azure-monitor/platform/metrics-supported#microsoftcomputevirtualmachinescalesets)

|                         |      |                                                                    |                                                  |  |
|-------------------------|------|--------------------------------------------------------------------|--------------------------------------------------|--|
| Metric                  | Unit | Description                                                        | *TODO replace this label with other information* |  |
| ----- ----- ----- ----- |      |                                                                    |                                                  |  |
|                         |      | Use this metric for <!-- put your specific information in here --> |                                                  |  |
|                         |      |                                                                    |                                                  |  |

```
<!-- Add additional explanation of reference information as needed here. Link to other articles such as your Monitor [servicename] article as appropriate. -->
```

```
<!-- Keep this text as-is -->
```

```
For more information, see a list of [all platform metrics supported in Azure Monitor]  
(https://docs.microsoft.com/azure/azure-monitor/platform/metrics-supported).
```

```
## Metric Dimensions
```

```
<!-- REQUIRED. Please keep headings in this order -->
```

```
<!-- If you have metrics with dimensions, outline it here. If you have no dimensions, say so. Questions  
email azmondocs@microsoft.com -->
```

```
For more information on what metric dimensions are, see [Multi-dimensional metrics](/azure/azure-monitor/platform/data-platform-metrics#multi-dimensional-metrics).
```

```
[TODO-replace-with-service-name] does not have any metrics that contain dimensions.
```

```
*OR*
```

```
[TODO-replace-with-service-name] has the following dimensions associated with its metrics.
```

```
<!-- See https://docs.microsoft.com/azure/storage/common/monitor-storage-reference#metrics-dimensions for an example. Part is copied below. -->
```

```
-----EXAMPLE format when you have dimensions-----
```

```
Azure Storage supports following dimensions for metrics in Azure Monitor.
```

| Dimension Name                            | Description                                                                                                                                                                                                                                                                                                                                                                                        |
|-------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| **BlobType**                              | The type of blob for Blob metrics only. The supported values are **BlockBlob**, **PageBlob**, and **Azure Data Lake Storage**. Append blobs are included in **BlockBlob**.                                                                                                                                                                                                                         |
| **BlobTier**                              | Azure storage offers different access tiers, which allow you to store blob object data in the most cost-effective manner. See more in [Azure Storage blob tier](/azure/storage/blobs/storage-blob-storage-tiers). The supported values include: <br/> <li>**Hot**: Hot tier</li> <li>**Cool**: Cool tier</li> <li>**Archive**: Archive tier</li> <li>**Premium**: Premium tier for block blob</li> |
| <li>**P4/P6/P10/P15/P20/P30/P40/P50/P60** | : Tier types for premium page blob</li> <li>**Standard**: Tier type for standard page Blob</li> <li>**Untiered**: Tier type for general purpose v1 storage account</li>                                                                                                                                                                                                                            |
| **GeoType**                               | Transaction from Primary or Secondary cluster. The available values include **Primary** and **Secondary**. It applies to Read Access Geo Redundant Storage(RA-GRS) when reading objects from secondary tenant.                                                                                                                                                                                     |

```
## Resource logs
```

```
<!-- REQUIRED. Please keep headings in this order -->
```

```
This section lists the types of resource logs you can collect for [TODO-replace-with-service-name].
```

```
<!-- List all the resource log types you can have and what they are for -->
```

```
For reference, see a list of [all resource logs category types supported in Azure Monitor](/azure/azure-monitor/platform/resource-logs-schema).
```

```
-----**OPTION 1 EXAMPLE** -----
```

```
<!-- OPTION 1 - Minimum - Link to relevant bookmarks in https://docs.microsoft.com/azure/azure-monitor/platform/resource-logs-categories, which is auto generated from the REST API. Not all resource log types metrics are published depending on whether your product group wants them to be. If the resource log is published, but category display names are wrong or missing, contact your PM and tell them to update them in the Azure Monitor "shoebox" manifest. If this article is missing resource logs that you and the PM know are available, both of you contact azmondocs@microsoft.com.
```

```
-->
```

```
<!-- Example format. There should be AT LEAST one Resource Provider/Resource Type here. -->
```

```
This section lists all the resource log category types collected for [TODO-replace-with-service-name].
```

|                                                                                                                             |
|-----------------------------------------------------------------------------------------------------------------------------|
| Resource Log Type   Resource Provider / Type Namespace<br/> and link to individual metrics                                  |
| ----- -----                                                                                                                 |
| Web Sites   [Microsoft.web/sites](/azure/azure-monitor/platform/resource-logs-categories#microsoftwebsites)                 |
| Web Site Slots   [Microsoft.web/sites/slots](/azure/azure-monitor/platform/resource-logs-categories#microsoftwebsitesslots) |

-----\*\*OPTION 2 EXAMPLE\*\* -----

<!-- OPTION 2 - Link to the resource logs as above, but work in extra information not found in the automated metric-supported reference article. NOTE: YOU WILL NOW HAVE TO MANUALLY MAINTAIN THIS SECTION to make sure it stays in sync with the resource-log-categories link. You can group these sections however you want provided you include the proper links back to resource-log-categories article.

-->

<!-- Example format. Add extra information -->

### Web Sites

Resource Provider and Type: [Microsoft.web/sites](/azure/azure-monitor/platform/resource-logs-categories#microsoftwebsites)

|                                                                                                       |
|-------------------------------------------------------------------------------------------------------|
| Category   Display Name   *TODO replace this label with other information*                            |
| ----- ----- -----                                                                                     |
| AppServiceAppLogs   App Service Application Logs   *TODO other important information about this type* |
| AppServiceAuditLogs   Access Audit Logs   *TODO other important information about this type*          |
| etc.                                                                                                  |

### Web Site Slots

Resource Provider and Type: [Microsoft.web/sites/slots](/azure/azure-monitor/platform/resource-logs-categories#microsoftwebsitesslots)

|                                                                                                       |
|-------------------------------------------------------------------------------------------------------|
| Category   Display Name   *TODO replace this label with other information*                            |
| ----- ----- -----                                                                                     |
| AppServiceAppLogs   App Service Application Logs   *TODO other important information about this type* |
| AppServiceAuditLogs   Access Audit Logs   *TODO other important information about this type*          |
| etc.                                                                                                  |

-----\*\*END Examples\*\* -----

## Azure Monitor Logs tables

<!-- REQUIRED. Please keep heading in this order -->

This section refers to all of the Azure Monitor Logs Kusto tables relevant to [TODO-replace-with-service-name] and available for query by Log Analytics.

-----\*\*OPTION 1 EXAMPLE\*\* -----

<!-- OPTION 1 - Minimum - Link to relevant bookmarks in https://docs.microsoft.com/azure/azure-monitor/reference/tables/tables-resourcetype where your service tables are listed. These files are auto generated from the REST API. If this article is missing tables that you and the PM know are available, both of you contact azmondocs@microsoft.com.

-->

<!-- Example format. There should be AT LEAST one Resource Provider/Resource Type here. -->

|                                                                                                                    |
|--------------------------------------------------------------------------------------------------------------------|
| Resource Type   Notes                                                                                              |
| ----- -----                                                                                                        |
| [Virtual Machines](/azure/azure-monitor/reference/tables/tables-resourcetype#virtual-machines)                     |
| [Virtual machine scale sets](/azure/azure-monitor/reference/tables/tables-resourcetype#virtual-machine-scale-sets) |

-----\*\*OPTION 2 EXAMPLE\*\* -----

<!-- OPTION 2 - List out your tables adding additional information on what each table is for. Individually link to each table using the table name. For example, link to [AzureMetrics] (<https://docs.microsoft.com/azure/azure-monitor/reference/tables/azurermetrics>).-->

NOTE: YOU WILL NOW HAVE TO MANUALLY MAINTAIN THIS SECTION to make sure it stays in sync with the automatically generated list. You can group these sections however you want provided you include the proper links back to the proper tables.

-->

### ### Virtual Machines

| Table                                                                | Description                                                                                                                                                                                        | *TODO replace this label with proper title for your additional information* |  |
|----------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|--|
| [AzureActivity](/azure/azure-monitor/reference/tables/azureactivity) | <!-- description copied from previous link --> Entries from the Azure Activity log that provides insight into any subscription-level or management group level events that have occurred in Azure. | *TODO other important information about this type                           |  |
| [AzureMetrics](/azure/azure-monitor/reference/tables/azurmetrics)    | <!-- description copied from previous link --> Metric data emitted by Azure services that measure their health and performance.                                                                    | *TODO other important information about this type                           |  |
| etc.                                                                 |                                                                                                                                                                                                    |                                                                             |  |

### ### Virtual Machine Scale Sets

| Table                                                                                          | Description                                                                                                                                                                                                                                         | *TODO replace this label with other information*  |  |
|------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|--|
| [ADAssessmentRecommendation](/azure/azure-monitor/reference/tables/adassessmentrecommendation) | <!-- description copied from previous link --> Recommendations generated by AD assessments that are started through a scheduled task. When you schedule the assessment it runs by default every 7 days and upload the data into Azure Log Analytics | *TODO other important information about this type |  |
| [ADReplicationResult](/azure/azure-monitor/reference/tables/adreplicationresult)               | <!-- description copied from previous link --> The AD Replication Status solution regularly monitors your Active Directory environment for any replication failures.                                                                                | *TODO other important information about this type |  |
| etc.                                                                                           |                                                                                                                                                                                                                                                     |                                                   |  |

<!-- Add extra information if required -->

For a reference of all Azure Monitor Logs / Log Analytics tables, see the [Azure Monitor Log Table Reference](/azure/azure-monitor/reference/tables/tables-resourcetype).

-----\*\*END EXAMPLES\*\* -----

### ### Diagnostics tables

<!-- REQUIRED. Please keep heading in this order -->

<!-- If your service uses the AzureDiagnostics table in Azure Monitor Logs / Log Analytics, list what fields you use and what they are for. Azure Diagnostics is over 500 columns wide with all services using the fields that are consistent across Azure Monitor and then adding extra ones just for themselves. If it uses service specific diagnostic table, refers to that table. If it uses both, put both types of information in. Most services in the future will have their own specific table. If you have questions, contact azmondocs@microsoft.com -->

[TODO-replace-with-service-name] uses the [Azure Diagnostics](/azure/azure-monitor/reference/tables/azurediagnostics) table and the [TODO whatever additional] table to store resource log information. The following columns are relevant.

\*\*Azure Diagnostics\*\*

| Property | Description |
|----------|-------------|
|          |             |
|          |             |
|          |             |

\*\*[TODO Service-specific table]\*\*

| Property | Description |
|----------|-------------|
|          |             |
|          |             |
|          |             |

## Activity log

<!-- REQUIRED. Please keep heading in this order -->

The following table lists the operations related to [TODO-replace-with-service-name] that may be created in the Activity log.

```
<!-- Fill in the table with the operations that can be created in the Activity log for the service. -->
operation	Description
```

<!-- NOTE: This information may be hard to find or not listed anywhere. Please ask your PM for at least an incomplete list of what type of messages could be written here. If you can't locate this, contact azmondocs@microsoft.com for help -->

For more information on the schema of Activity Log entries, see [Activity Log schema](/azure/azure-monitor/essentials/activity-log-schema).

## Schemas

<!-- REQUIRED. Please keep heading in this order -->

The following schemas are in use by [TODO-replace-with-service-name]

<!-- List the schema and their usage. This can be for resource logs, alerts, event hub formats, etc depending on what you think is important. -->

## See Also

<!-- replace below with the proper link to your main monitoring service article -->

- See [Monitoring Azure [TODO-replace-with-service-name]](monitor-service-name.md) for a description of monitoring Azure [TODO-replace-with-service-name].
- See [Monitoring Azure resources with Azure Monitor](/azure/azure-monitor/insights/monitor-azure-resources) for details on monitoring Azure resources.

# .NET template for a tutorial article

5/10/2021 • 5 minutes to read

This article provides a Markdown template that you should use when writing a .NET [tutorial article](#).

All articles need the [metadata header](#) along with the required values for reporting. Detailed instructions for all tutorial articles are available in the [Tutorial template article](#).

## Standard Metadata values for .NET areas

The .NET Docs team uses folder-based metadata where ever possible. You specify fewer metadata fields in your tutorial. Using hackable URLs, and placing your article in the correct directory, you can omit **ms.prod**, **ms.technology**, and **ms.devlang** from the metadata block. Consult [.openpublishing.publish.configjson](#) on the dotnet/docs repo for the current configuration.

## Markdown template

You can copy and paste the following Markdown for your tutorial article:

```
---
title: Intent and product brand in a unique string of 43-59 chars including spaces - do not include site
identifier (it is auto-generated). Include the word "tutorial".
description: 115-145 characters including spaces. Edit the intro para describing article intent to fit here.
This abstract displays in the search result.
author: github-alias
ms.author: MSFT-alias-person-or-DL
ms.date: 12/05/2017
ms.topic: tutorial
---

<!--Recommended: Remove all the comments in this template before you
sign-off or merge to master.-->

<!--Tutorials are scenario-based procedures for the top customer tasks
identified in milestone one of the
[Content & Learning content model](contribute-get-started-mvc.md).
You only use tutorials to show the single best procedure for completing
an approved top 10 customer task.
-->

# Tutorial: <do something with X>
<!--Required:
Starts with "Tutorial:"
Make the first word following "Tutorial:" a verb.
-->

Introductory paragraph.
<!--Required:
Lead with a light intro that describes, in customer-friendly language,
what the customer will learn, or do, or accomplish. Answer the
fundamental "why would I want to do this?" question.
-->

In this tutorial, you learn how to:

> [<div class="checklist">]
> * All tutorials include a list summarizing the steps to completion
> * Each of these bullet points align to a key H2
> * Use these green checkboxes in a tutorial
<!--Required:
```

The outline of the tutorial should be included in the beginning and at the end of every tutorial. These will align to the \*\*procedural\*\* H2 headings for the activity. You do not need to include all H2 headings. Leave out the prerequisites, clean-up resources and next steps--->

If you don't have a <service> subscription, create a free trial account...  
<!-- Required, if a free trial account exists  
Because tutorials are intended to help new customers use the product or service to complete a top task, include a link to a free trial before the first H2, if one exists. You can find listed examples in [Write tutorials](contribute-how-to-mvc-tutorial.md)  
--->

<!--Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.-->

## ## Prerequisites

- First prerequisite
- Second prerequisite
- Third prerequisite

<!--If you need them, make Prerequisites your first H2 in a tutorial. If there's something a customer needs to take care of before they start (for example, creating a VM) it's OK to link to that content before they begin.-->

## ## Procedure 1

<!--Required:  
Tutorials are prescriptive and guide the customer through an end-to-end procedure. Make sure to use specific naming for setting up accounts and configuring technology.

Don't link off to other content - include whatever the customer needs to complete the scenario in the article. For example, if the customer needs to set permissions, include the permissions they need to set, and the specific settings in the tutorial procedure. Don't send the customer to another article to read about it.

--->

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure
  - ![Browser](media/contribute-how-to-mvc-tutorial/browser.png)

<!--Use screenshots but be judicious to maintain a reasonable length.  
Make sure screenshots align to the [current standards](https://review.docs.microsoft.com/help/contribute/contribute-how-to-create-screenshot?branch=master).  
If users access your product/service via a web browser the first screenshot should always include the full browser window in Chrome or Safari. This is to show users that the portal is browser-based - OS and browser agnostic.-->
1. Step four of the procedure

## ## Procedure 2

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure

## ## Procedure 3

Include a sentence or two to explain only what is needed to complete the procedure.

<!--Code requires specific formatting. Here are a few useful examples of

.. code snippets for marking up code samples or commonly used code blocks.

Code should be included from buildable samples, as shown below:

```
[!code-<language>[<name>](<pathToFile><queryoption><queryoptionvalue>)]  
* `-<language>` (*optional* but *recommended*)  
* Language of the code snippet being referenced.  
  
* `<name>` (*optional*)  
* Name for the code snippet. It doesn't have any impact on the output HTML,  
but you can use it to improve the readability of your Markdown source.  
  
* `<pathToFile>` (*mandatory*)  
* Relative path in the file system that indicates the code snippet file to reference.  
  
* `<queryoption>` and `<queryoptionvalue>` (*optional*)  
* Used together to specify how the code should be retrieved from the file:  
* `#: `#L{startlinenumber}-L{endlinenumber}` (line range) *or* `#{tagname}` (tag name).  
We discourage the use of line numbers because they are very brittle. Tag name is the preferred way of  
referencing code snippets.  
* `range`: `?range=1,3-5` A range of lines. This example includes lines 1, 3, 4, and 5.  
* `dedent`: `?dedent=8` Dedents the lines by a number of spaces--in this case, 8. This can be combined  
with  
the `range` and other query options that select a subset of the lines of a file.  
* `outdent`: `?outdent=8` Reverses the indent of the lines by a number of spaces--in this case, 8. This  
can  
be combined with `range` and other query options that select a subset of the lines of a file.
```

We recommend using the tag name option whenever possible. The tag name is the name of a region or of a code comment

in the format of `Snippetttagname` present in the source code. The following example shows how to refer to the tag

name `ThrowAll`:

```
[!code-csharp[csrefKeyword#1](../../../../samples/snippets/csharp/language-reference/keywords/throw/throw-1.cs#ThrowAll)]
```

--->

## Get the code

Provide a link to the finished tutorial in the <https://github.com/dotnet/samples> repository. Link to the root directory of the sample.

Provide a second link to a zip file in the parent directory of the sample.

## Additional resources

Provide links to where you can learn more about the topics covered in this tutorial

## Next steps

Advance to the next article to learn how to create...

```
> [&gt;div class="nextstepaction"]  
> [Next steps button](contribute-get-started-mvc.md)
```

<!-- Required:

Tutorials should always have a Next steps H2 that points to the next logical tutorial in a series, or, if there are no other tutorials, to some other cool thing the customer can do. A single link in the blue box format should direct the customer to the next article - and you can shorten the title in the boxes if the original one doesn't fit.

Do not use a "More info section" or a "Resources section" or a "See also section". --->

# .NET template for an advanced tutorial

4/16/2021 • 6 minutes to read

This article provides a Markdown template that you should use when writing a .NET or ASP.NET advanced tutorial. Here are some guidelines for choosing the tutorial or advanced tutorial template:

- Tutorial:
  - Relatively short.
  - Purpose is to help the customer complete a top task as quickly as possible. The achievement may spark interest in going deeper or broader, but the tutorial itself isn't designed to provide that kind of coverage.
- Advanced tutorial:
  - Relatively long, or a series.
  - Purpose is to cover a broader range of topics, to take more time to better prepare the customer to use the service(s) or feature(s) for tasks of real-world complexity.
  - Could be follow-up to a tutorial or could start out with the basics.

All articles need the [metadata header](#) along with the required values for reporting. Detailed instructions for all tutorial articles are available in the [Tutorial template article](#).

## Standard Metadata values for .NET areas

The .NET and ASP.NET Docs team uses folder-based metadata wherever possible. You specify fewer metadata fields in your tutorial. Using hackable URLs, and placing your article in the correct directory, you can omit `ms.prod`, `ms.technology`, and `ms.devlang` from the metadata block. Consult [.openpublishing.publish.config.json](#) on the dotnet/docs or aspnet/docs repo for the current configuration.

## Markdown template

You can copy and paste the following Markdown for your tutorial article:

```
---
title: Intent and product brand in a unique string of 43-59 chars including spaces - do not include site
identifier (it is auto-generated.)
description: 115-145 characters including spaces. Edit the intro para describing article intent to fit here.
This abstract displays in the search result.
author: github-alias
ms.author: MSFT-alias-person-or-DL
ms.date: 12/05/2017
ms.topic: tutorial
---

<!--Recommended: Remove all the comments in this template before you
sign-off or merge to master.-->

<!--Tutorials are scenario-based procedures for customer tasks.
-->

# Tutorial: <do something with X>
<!--Required:
Starts with "Tutorial:
Make the first word following "Tutorial:" a verb.
-->

# Introductory paragraph
```

introduction paragraph.

<!--Required:

Lead with a light intro that describes, in customer-friendly language, what the customer will learn, or do, or accomplish. Answer the fundamental “why would I want to do this?” question. Mention that this tutorial is relatively long and complex.

-->

In this tutorial, you learn how to:

```
> [&gt;div class="checklist"]
```

```
> * All tutorials include a list summarizing the steps to completion
```

```
> * Each of these bullet points align to a key H2
```

```
> * Use these green checkboxes in a tutorial
```

<!--Required:

The outline of the tutorial should be included in the beginning of every tutorial. Most of these will align to the \*\*procedural\*\* H2 headings for the activity. Leave out Prerequisites, Clean up resources, and Next steps, and any others as needed.

-->

If you don't have a <service> subscription, create a free trial account...

<!-- Required, if a free trial account exists

Because tutorials are intended to help new customers use the product or service to complete a top task, include a link to a free trial before the first H2, if one exists. You can find listed examples in

[Write tutorials](contribute-how-to-mvc-tutorial.md)

-->

<!--Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.-->

## Prerequisites

- First prerequisite
- Second prerequisite
- Third prerequisite

<!--If you need them, make Prerequisites your first H2 in a tutorial. If there's something a customer needs to take care of before they start (for example, creating a VM) it's OK to link to that content before they begin.-->

## Sign in to <service/product/tool name>

Sign in to the [<service> portal](url).

<!--If you need to sign in to the portal to do the tutorial, this H2 and link are required.-->

## Procedure 1

<!--Required:

Tutorials are prescriptive and guide the customer through an end-to-end procedure. Make sure to use specific naming for setting up accounts and configuring technology.

Provide customers with everything they need to know in the tutorial to successfully complete the tutorial. For example, if the customer needs to set permissions, include the permissions they need to set, and the specific settings in the tutorial procedure. Don't send the customer to another article to find what they need to understand what they're doing. Links should only be provided for when the customer wants to investigate a subject in greater depth.

The scope should be no more nor less than is needed for the tasks the tutorial is showing how to accomplish. The “advanced tutorial” doc type is not a license to include everything the writer knows on the subject. The tutorial should lead the user through a sequence of steps with minimal digressions, providing only information directly relevant to the task at hand.

-->

Include a sentence or two to explain what is needed to complete the procedure. Include background information or reason why explanations as appropriate.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure

![Browser](media/contribute-how-to-mvc-tutorial/browser.png)  
!---Use screenshots only where needed for understanding.  
Make sure screenshots align to the  
[current standards](https://review.docs.microsoft.com/help/contribute/contribute-how-to-create-screenshot?branch=master).  
If users access your product/service via a web browser the first screenshot should always include the full browser window in Chrome or Safari. This is to show users that the portal is browser-based - OS and browser agnostic.--->
1. Step four of the procedure

## Procedure 2

Include a sentence or two to explain what is needed to complete the procedure. Include background information or reason why explanations as appropriate.

## Procedure 3

Include a sentence or two to explain only what is needed to complete the procedure. Include background information or reason why explanations as appropriate.

<!--Code requires specific formatting. Here are a few useful examples of commonly used code blocks.

Code should be included from buildable samples, as shown below:

```
[!code-<language>[<name>](<pathToFile><queryoption><queryoptionvalue>)]  
  
* `--<language>` (*optional* but *recommended*)  
  * Language of the code snippet being referenced.  
  
* `<name>` (*optional*)  
  * Name for the code snippet. It doesn't have any impact on the output HTML,  
    but you can use it to improve the readability of your Markdown source.  
  
* `<pathToFile>` (*mandatory*)  
  * Relative path in the file system that indicates the code snippet file to reference.  
  
* `<queryoption>` and `<queryoptionvalue>` (*optional*)  
  * Used together to specify how the code should be retrieved from the file:  
    * `#: `#`{startlinenumber}-L{endlinenumber}` (line range) *or* `#{tagname}` (tag name).  
    We discourage the use of line numbers because they are very brittle. Tag name is the preferred way of referencing code snippets.  
    * `range`: `?range=1,3-5` A range of lines. This example includes lines 1, 3, 4, and 5.  
    * `dedent`: `?dedent=8` Dedents the lines by a number of spaces--in this case, 8. This can be combined with  
      the `range` and other query options that select a subset of the lines of a file.  
    * `outdent`: `?outdent=8` Reverses the indent of the lines by a number of spaces--in this case, 8. This can  
      be combined with `range` and other query options that select a subset of the lines of a file.
```

We recommend using the tag name option whenever possible. The tag name is the name of a region or of a code comment in the format of `Snippettagname` present in the source code. The following example shows how to refer to the tag name `ThrowAll`:

```
[!code-csharp[csrefKeyword#1](../../../../samples/snippets/csharp/language-reference/keywords/throw/throw-1.cs#ThrowAll)]
```

```
-->

## Get the code

Provide a link to the finished tutorial in the https://github.com/dotnet/samples repository. Link to the root directory of the sample.

Provide a second link to a zip file in the parent directory of the sample.

## Additional resources

Provide links to where you can learn more about the topics covered in this tutorial

## Next steps

Advance to the next article to learn how to create...
> [<div class="nextstepaction">
> [Next steps button](contribute-get-started-mvc.md)

<!-- Required:
Tutorials should always have a Next steps H2 that points to the next
logical tutorial in a series, or, if there are no other tutorials, to
some other cool thing the customer can do. A single link in the blue box
format should direct the customer to the next article - and you can
shorten the title in the boxes if the original one doesn't fit.
Do not use a "More info section" or a "Resources section" or a "See also
section". -->
```

# .NET template for a quickstart article

5/10/2021 • 8 minutes to read

This article provides a Markdown template that you should use when writing a .NET [quickstart article](#).

All articles need the [metadata header](#) along with the required values for reporting. Detailed instructions for all tutorial articles are available in the [Tutorial template article](#).

Content should be brief. Don't require prerequisites or lengthy procedures. Don't include an outline of the steps with green checkboxes.

## Standard .NET Metadata values

The .NET Docs team uses folder-based metadata where ever possible. You specify fewer metadata fields in your tutorial. Using hackable URLs, and placing your article in the correct directory, you can omit **ms.prod**, **ms.technology**, and **ms.devlang** from the metadata block. Consult [.openpublishing.publish.config.json](#) on the dotnet/docs repo for the current configuration.

## Interactive Quickstarts

You can copy and paste the following YAML for your interactive quickstart article.

```
### YamlMime:Tutorial
title: Hello C# - Interactive Quickstart
metadata:
    title: Hello World interactive tutorial - C# quickstarts
    description: In this quickstart, you'll use your browser to learn C# interactively. You write C# code and see the results of compiling and running your code directly in the browser.
    audience: Developer
    level: Beginner
    ms.topic: quickstart
    nextTutorialHref: link-to-next-step.yml
    nextTutorialTitle: Title for next quickstart
    displayType: two-column
    interactive: csharp
    ms.custom: mvc
items:
- durationInMinutes: 1
  content: |
    Introduction to the lessons in this quickstart.
    <!---interactive quickstarts are fundamental day-1 instructions for helping new customers use a library or language feature.
    You only use quickstarts when you can get the library, language feature, or functionality into the hands of new customers in less than 10 minutes using the current features of the interactive C# REPL. -->
    <!--Required:
    Lead with a light intro that describes, in customer-friendly language, what the customer will learn, or do, or accomplish. Answer the fundamental "why would I want to do this?" question.
    -->

- title: Each step should have a title that starts with a verb and explains what the reader will do and learn
  durationInMinutes: 2
  content: |
    Run the following code in the interactive window. To do that, type the following code block in the interactive window and click the **Run** button:
```

```
csharp
Console.WriteLine("Hello World!");
```
```

This should be less than a page. One step, or two at most. Note that the `interactive` suffix is not included. The site displays the interactive window in its own pane. Strive for at least 50% code on each page. Make sure that code is "above the fold" as you create the page, don't start with a wall of explanatory text.

```
> [!NOTE]
> This online coding experience is in preview mode. If you encounter problems, please report them [on the dotnet/try repo](https://github.com/dotnet/try/issues).
```

- title: Have a few sections and finish with a challenge
- durationInMinutes: 3
- content: |
  - Note that the code written by the reader will persist as they navigate through the process.
  - Ask them to try a few things and experiment on each page.

- title: Search strings
- durationInMinutes: 10
- content: |
  - Your last page should end with a challenge. An example follows:

\*\*\*Challenge\*\*\*

There are two similar methods, `<xref:System.String.StartsWith%2A>` and `<xref:System.String.EndsWith%2A>` that also search for sub-strings in a string. These find a substring at the beginning or the end of the string. Try to modify the previous sample to use `<xref:System.String.StartsWith%2A>` and `<xref:System.String.EndsWith%2A>` instead of `<xref:System.String.Contains%2A>`. Search for "You" or "goodbye" at the beginning of a string. Search for "hello" or "goodbye" at the end of a string.

```
> [!Note]
> Watch your punctuation when you test for the text at the end of the string. If the string
> ends with a period, you must check for a string that ends with a period.
```

You should get `true` for starting with "You" and ending with "hello" and false for starting with or ending with "goodbye".

```
> [!NOTE]
> This online coding experience is in preview mode. If you encounter problems, please report them [on the dotnet/try repo](https://github.com/dotnet/try/issues).
```

- title: Complete challenge
- durationInMinutes: 3
- content: |
  - Show one possible answer on the next page.

Did you come up with something like this?

```
```csharp
string songLyrics = "You say goodbye, and I say hello";
Console.WriteLine(songLyrics.StartsWith("You"));
Console.WriteLine(songLyrics.StartsWith("goodbye"));

Console.WriteLine(songLyrics.EndsWith("hello"));
Console.WriteLine(songLyrics.EndsWith("goodbye"));
```
```

```
> [!NOTE]
> This online coding experience is in preview mode. If you encounter problems, please report them [on the dotnet/try repo](https://github.com/dotnet/try/issues).
```

- content: |
  - Tell them they've completed the quick start. Give them more to learn. Note that you do not need a link to the next quickstart. That's provided in the metadata above.
  - <!-- Required:
  - Quickstarts should always have a Next steps H2 that points to the next logical quickstart in a series, or, if there are no other quickstarts, to some other

cool thing the customer can do. A single link in the blue box format should direct the customer to the next article - and you can shorten the title in the boxes if the original one doesn't fit.  
Do not use a "More info section" or a "Resources section" or a "See also section". --->

You can learn more about working with the `string` type in the [C# Programming Guide](../programming-guide/index.md) article on [strings](../programming-guide/strings/index.md).  
[How to tips on working with strings](../how-to/index.md#working-with-strings).

## Local quickstarts

```
---
title: Page title has the greatest impact on search
description: In this quickstart you will learn how to...
ms.date: 03/05/2018
ms.custom: mvc
---
<!--Recommended: Remove all the comments in this template before you sign-off or merge to master.-->
<!--quickstarts are fundamental day-1 instructions for helping new customers use a subscription to quickly try out a specific product/service. The entire activity is a short set of steps that provides an initial experience.
You only use quickstarts when you can get the service, technology, or functionality into the hands of new customers in less than 10 minutes.
-->
```

```
# Quickstart: <do something with X>
<!--Required:
Starts with "quickstart: "
Make the first word following "quickstart:" a verb.
-->
```

```
Introductory paragraph.
<!--Required:
Lead with a light intro that describes, in customer-friendly language, what the customer will learn, or do, or accomplish. Answer the fundamental "why would I want to do this?" question.
-->
```

```
In this quickstart, you will <do X>

Create an introduction that briefly describes what you are going to create in 1-2 sentences. The entire activity is a short set of steps that provide an initial experience in less than 10 minutes. The quickstart should target new customers to this technology.
```

```
This article explains how to do something brief. Make sure to include information about downloads or portal access that could be worded as follows:
```

```
<!--Avoid notes, tips, and important boxes. Readers tend to skip over them.
Better to put that info directly into the article text.-->
```

```
## Prerequisites
```

```
This quickstart expects you to have a machine you can use for development. The .NET article [Get Started in 10 minutes](https://www.microsoft.com/net/core) has instructions for setting up your local development environment on Mac, PC or Linux. A quick overview of the commands you'll use is in the [introduction to the local quickstarts](local-environment.md) with links to more details.
```

```
-->
```

```
<!--Required:  
Quickstarts are prescriptive and guide the customer through an end-to-end  
procedure. Make sure to use specific naming for setting up accounts and  
configuring technology.  
Don't link off to other content - include whatever the customer needs to  
complete the scenario in the article. For example, if the customer needs  
to set permissions, include the permissions they need to set, and the specific  
settings in the quickstart procedure. Don't send the customer to another  
article to read about it.  
In a break from tradition, do not link to reference topics in the procedural  
part of the quickstart when using cmdlets or code. Provide customers what they  
need to know in the quickstart to successfully complete the quickstart.  
For portal-based procedures, minimize bullets and numbering.  
For the CLI or PowerShell based procedures, don't use bullets or numbering.  
-->
```

Include a sentence or two to explain only what is needed to complete the procedure.

```
1. Step one of the procedure  
1. Step two of the procedure  
1. Step three of the procedure  
![Browser](media/contribute-how-to-mvc-quickstart/browser.png)  
<!--Use screenshots but be judicious to maintain a reasonable length. Make  
sure screenshots align to the  
[current standards](https://review.docs.microsoft.com/help/contribute/contribute-how-to-create-  
screenshot?branch=master).  
If users access your product/service via a web browser the first screenshot  
should always include the full browser window in Chrome or Safari. This is  
to show users that the portal is browser-based - OS and browser agnostic.-->  
1. Step four of the procedure
```

```
## Procedure 1
```

At least one procedural step is required. The H2 should begin with a verb,  
such as 'Create' and include what the reader will learn or do.

```
## Procedure 2
```

Include a sentence or two to explain only what is needed to complete the procedure.

```
1. Step one of the procedure  
1. Step two of the procedure  
1. Step three of the procedure
```

```
## Procedure 3
```

Code blocks must be part of technical content. Make sure to use the interactive  
functionality where possible.

Here is an example of a code block for C#:

```
```csharp-interactive  
int a = 18;  
int b = 6;  
int c = a + b;  
Console.WriteLine(c);  
```
```

```
## Add a challenge
```

After explaining the basics, close with a challenge: Give the readers a problem  
and ask them to solve it.

Give them a link to check their work, such as:

You can check your answer by [looking at the finished sample code on GitHub]  
(<https://github.com/dotnet/samples/tree/master/csharp/numbers-quickstart/Program.cs#L104-L106>)

```
## Next steps

## Next steps

Include a brief sentence to entice readers to advance to the next article to
learn more.
> [&gt; [!div class="nextstepaction"]
> [Next steps button](../../contribute-get-started-mvc.md)

<!-- Required:
Quickstarts should always have a Next steps H2 that points to the next logical
quickstart in a series, or, if there are no other quickstarts, to some other
cool thing the customer can do. A single link in the blue box format should
direct the customer to the next article - and you can shorten the title in the
boxes if the original one doesn't fit.
Do not use a "More info section" or a "Resources section" or a "See also section". -->
```

# SQL template for a tutorial article

5/10/2021 • 4 minutes to read

This article provides a Markdown template that you should use when writing a SQL [tutorial article](#).

All articles need the [metadata header](#) along with the required values for reporting. Detailed instructions for completing this template are available in the [Contributor Guide](#).

You can copy and paste the following Markdown for your tutorial article:

```
---
title: Intent and product brand in a unique string of 43-59 chars including spaces - do not include site
identifier (it is auto-generated.)
description: 115-145 characters including spaces. Edit the intro para describing article intent to fit here.
This abstract displays in the search result.
author: github-alias
ms.author: MSFT-alias-person-or-DL
ms.date: 12/05/2017
ms.topic: tutorial
ms.prod: sql-non-specified|analysis-services|reporting-services
ms.prod_service: "sql"
ms.technology: from allowlist
ms.component: from allowlist

# Optional fields:
# monikerRange: For versioning (https://aka.ms/sqlversioning-msft)
# ms.custom:
# manager: MSFT-alias-manager-or-PM-counterpart
---

<!--Recommended: Remove all the comments in this template before you
sign-off or merge to master.-->

<!--Tutorials are scenario-based procedures for the top customer tasks
identified in milestone one of the
[Content & Learning content model](contribute-get-started-mvc.md).
You only use tutorials to show the single best procedure for completing
an approved top 10 customer task.
-->

# Tutorial: <do something with X>
<!--Required:
Starts with "Tutorial:"
Make the first word following "Tutorial:" a verb.
-->

<!-- Applies-to includes specify which SQL products this article
applies to. For guidance on how to find and use them see
http://aka.ms/editsqldocs-msft#applies-to-includes. Make sure to
use the correct relative path to the includes directory -->
[!INCLUDE[name-of-applies-to-include](./includes/name-of-applies-to-include.md)]>

Introductory paragraph.
<!--Required:
Lead with a light intro that describes, in customer-friendly language,
what the customer will learn, or do, or accomplish. Answer the
fundamental "why would I want to do this?" question.
-->

In this tutorial, you learn how to:
> [!div class="checklist"]
```

```
> * All tutorials include a list summarizing the steps to completion
> * Each of these bullet points align to a key H2
> * Use these green checkboxes in a tutorial
<!--Required:
The outline of the tutorial should be included in the beginning and at
the end of every tutorial. These will align to the **procedural** H2
headings for the activity. You do not need to include all H2 headings.
Leave out the prerequisites, clean-up resources and next steps-->

<!--Avoid notes, tips, and important boxes. Readers tend to skip over
them. Better to put that info directly into the article text.-->
```

## ## Prerequisites

- First prerequisite
- Second prerequisite
- Third prerequisite

<!--If you need them, make Prerequisites your first H2 in a tutorial. If there's something a customer needs to take care of before they start (for example, creating a VM) it's OK to link to that content before they begin.-->

## ## Procedure 1

### <!--Required:

Tutorials are prescriptive and guide the customer through an end-to-end procedure. Make sure to use specific naming for setting up accounts and configuring technology.

Don't link off to other content - include whatever the customer needs to complete the scenario in the article. For example, if the customer needs to set permissions, include the permissions they need to set, and the specific settings in the tutorial procedure. Don't send the customer to another article to read about it.

In a break from tradition, do not link to reference topics in the procedural part of the tutorial when using cmdlets or code. Provide customers what they need to know in the tutorial to successfully complete the tutorial.

-->

No less than 2 procedural sections, and no more than 12 steps across the procedural sections

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure
  - ![Browser](media/contribute-how-to-mvc-tutorial/browser.png)

<!--Use screenshots but be judicious to maintain a reasonable length.  
Make sure screenshots align to the [current standards](https://review.docs.microsoft.com/help/contribute/contribute-how-to-create-screenshot?branch=master).

If users access your product/service via a web browser the first screenshot should always include the full browser window in Chrome or Safari. This is to show users that the portal is browser-based - OS and browser agnostic.-->
1. Step four of the procedure

## ## Procedure 2

Include a sentence or two to explain only what is needed to complete the procedure.

1. Step one of the procedure
1. Step two of the procedure
1. Step three of the procedure

## ## Clean up resources

Optional section if there are resources that can be cleaned up.

If you're not going to continue to use this application, delete <resources> with the following steps:

1. From the left-hand menu...
2. ...click Delete, type...and then click Delete

<!--Required:

To avoid any costs associated with following the tutorial procedure, a Clean up resources (H2) should come just before Next steps (H2)

-->

## Additional resources

Optional section. Provide links to where you can learn more about the topics covered in this tutorial

## Next steps

Advance to the next article to learn how to create...

- > [&!div class="nextstepaction"]
- > [Next steps button](contribute-get-started-mvc.md)

<!-- Required:

Tutorials should always have a Next steps H2 that points to the next logical tutorial in a series, or, if there are no other tutorials, to some other cool thing the customer can do. A single link in the blue box format should direct the customer to the next article - and you can shorten the title in the boxes if the original one doesn't fit.

Do not use a "More info section" or a "Resources section" or a "See also section". -->

# Base template for general troubleshooting articles

5/10/2021 • 3 minutes to read

This article provides a general troubleshooting template in markdown code that you should use when writing a [general troubleshooting article](#).

All articles need the [metadata header](#) along with the required values for reporting.

## TIP

This template is available in the [Docs Authoring Pack for VS Code](#). To automatically create a new Markdown file with skeleton content and guidance, select **Template** from the Docs Markdown menu in VS Code, then select the appropriate template from the dropdown list. For more information, see [How to use Docs templates](#).

You can copy and paste the following Markdown for your general troubleshooting article:

```
---
title: #Required; page title displayed in search results. You should include "troubleshooting" in the title and description.
description: #Required; article description that is displayed in search results.
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; Microsoft alias of author; optional team alias.
ms.topic: troubleshooting #Required
ms.date: #Required; mm/dd/yyyy format.
---

<!--General troubleshooting articles are written when a specific error message isn't known. The customer has encountered an issue that needs to be resolved without being clear about what is causing the issue.
-->

# Troubleshooting <some feature>
<!--Required:
Include the word troubleshooting.
-->

Introductory paragraph.
<!--Required:
Lead with a light intro that describes, in customer-friendly language, what the customer should expect to see in the article. The information in the introduction should help the customer decide whether the information applies to the issue that they are encountering.
-->

<!--Avoid notes, tips, and important boxes. Readers tend to skip over them. Better to put that info directly into the article text.-->

## Prerequisites
<!-- If you need them, make Prerequisites your first H2 in the article. If there's something a customer needs to take care of before they start it's OK to link to that content before they begin.
-->

- First prerequisite
- Second prerequisite
- Third prerequisite

## Potential quick fixes
```

<!-- An issue might be able to be temporarily resolved with a quick fix.  
If known, list any solutions that can be implemented quickly to resolve  
the issue. Link to information about a longer term solution in the root  
cause and solutions section.

--->

### Fix 1

1. Step 1.  
2. Step 2.

### Fix 2

1. Step 1.  
2. Step 2.

## Root cause and solutions

<!-- To be able to identify the issue and how to prevent it from happening again, the cause of the issue  
should be defined if known. Make sure that the H3 headings clearly state the intention of the section. Each  
section should have a short sentence that describes the steps that are about to be taken.

--->

Root cause.

### Solution 1

<!-- Make sure that the H3 headings clearly state the intention of the section. Each section should have a  
short sentence that describes the steps that are about to be taken.

--->

Solution summary.

1. Step 1.  
2. Step 2.

### Solution 2

<!-- Make sure that the H3 headings clearly state the intention of the section. Each section should have a  
short sentence that describes the steps that are about to be taken.

--->

Solution summary.

1. Step 1.  
2. Step 2.

## Advanced troubleshooting and data collection

<!-- Include this section if the issue requires advanced troubleshooting steps that may require a call to  
support, list any information or procedures in this section to help the customer prepare for submitting a  
support ticket.

--->

## Next steps

<!-- If there are any next steps that should be taken after the issue has been initially resolved, list  
them in this section.

--->

- Next step 1  
- Next step 2

# Base template for known issues articles

5/10/2021 • 2 minutes to read

This article provides a known issues template in markdown code that you should use when writing a [known issues article](#).

All articles need the [metadata header](#) along with the required values for reporting.

You can copy and paste the following Markdown for your known issues article:

```
---
```

```
title: Known issues - feature-name #Required; Change feature-name to the name of the feature or service that the known issues relate to.
```

```
description: #Required; article description that is displayed in search results. Include the complete message that the customer sees.
```

```
author: #Required; your GitHub user alias, with correct capitalization.
```

```
ms.author: #Required; microsoft alias of author; optional team alias.
```

```
ms.topic: troubleshooting #Required
```

```
ms.date: #Required; mm/dd/yyyy format.
```

```
-->
```

```
<!--Recommended: Remove all the comments in this template before you sign-off or merge to master.-->
```

```
<!-- Known issues articles help inform customers of issues that they may encounter that are currently being worked on or planned to be fixed in the near future. Known issues added to the article should be removed when the issue has been resolved. If the issue is never going to be resolved, it should be documented in a conceptual or how-to article as expected behavior.
```

```
-->
```

```
# Known issues: <feature name>
```

```
<!--Required-->
```

```
This article provides information about known issues associated with <feature-name>.
```

```
## <Issue title>
```

```
<!--Required:
```

```
Each known issue should be in its own section. Provide a title for the section that enables the customer to easily identify the issue that they are experiencing.
```

```
-->
```

```
### Prerequisites
```

```
<!--Optional:
```

```
If there are steps that the customer should complete or tools that need to be downloaded before continuing through the troubleshooting guidance, they should be described in this section.
```

```
-->
```

```
- Prerequisite 1
```

```
- Prerequisite 2
```

```
### Troubleshooting steps
```

```
<!--Optional:
```

```
Not all known issues will be correctable, but if so, add this section describing the steps to take to correct the issue.
```

```
-->
```

```
1. Step 1.
```

```
2. Step 2.
```

```
### Possible causes
```

```
<!--Required:
```

```
List known possible causes of the issue.
```

```
-->  
#### Cause #  
<!--Optional:  
Most common cause.  
-->  
#### Cause #  
<!--Optional:  
Next most common cause.  
-->  
  
## Next steps  
  
<!-- Optional:  
Include this section if there are 1 -3 concrete, highly relevant next steps the user should take. Delete if  
there are no next steps. This is not a place for a list of links. If you include links to next steps, make  
sure to include text to explain why the next steps are relevant or important. -->
```

# Base template for problem resolution articles

5/10/2021 • 2 minutes to read

This article provides a problem resolution template in markdown code that you should use when writing a [problem resolution article](#).

All articles need the [metadata header](#) along with the required values for reporting.

You can copy and paste the following Markdown for your problem resolution article:

```
---
title: #Required; page title displayed in search results. If short enough, include the message, or key words from the message.
description: #Required; article description that is displayed in search results. Include the complete message that the customer sees.
author: #Required; your GitHub user alias, with correct capitalization.
ms.author: #Required; microsoft alias of author; optional team alias.
ms.topic: troubleshooting #Required
ms.date: #Required; mm/dd/yyyy format.
---

<!--Recommended: Remove all the comments in this template before you sign-off or merge to master.-->

<!--Problem resolution articles help customers quickly identify the problem or error that they are having with a service or feature, identify the cause of the problem, and find steps that can be performed to resolve the problem.
-->

# <Problem message or key words>
<!--Required-->

Summary paragraph.
<!--Required:
The summary describes the symptom that the customer is experiencing. This summary should also include any prerequisite actions that should be performed before resolving the problem.
-->

## Symptoms

<!--Required:
Precisely describe what the customer should be experiencing when encountering the problem. If the title can't contain the complete message, expand on it here. If there is relevant general troubleshooting information available, link to it from here.
-->

## Cause
<!--Required:
Describe the cause of the symptoms. It is possible that there could be several causes for a problem. List each one as an H3 with **Cause #** where ***#*** is a successive number of possible causes.
-->
### Cause #
<!--Optional:
Most common cause.
-->
### Cause #
<!--Optional:
Next most common cause.
-->

## Solution
<!--Required:
```

List the steps that should be taken to resolve the problem. It is possible that there could be several solutions for a problem. If there are multiple solutions, put them in the order of complexity and provide instructions on how to choose from among them. If workaround information is available to temporarily alleviate the symptoms, list them in this section.

--->

### Solution #

<!--Optional:

Most common solution.

--->

### Solution #

<!--Optional:

Next most common solution.

--->

## Next steps

<!-- Optional:

Include this section if there are 1 -3 concrete, highly relevant next steps the user should take. Delete if there are no next steps. This is not a place for a list of links. If you include links to next steps, make sure to include text to explain why the next steps are relevant or important. --->

# What can I publish on docs.microsoft.com?

5/12/2021 • 10 minutes to read

Thank you for your interest in publishing to docs.microsoft.com. The site is designed as the single site for technical documentation for all Microsoft products and services.

For quality and consistency, we need to make sure that content stays within the boundaries of technical documentation.

## Acceptable technical content

The following types of content are accepted in docs.microsoft.com repositories. They are authored in our private and public repos and then published to <https://docs.microsoft.com/>.

**Technical articles about using the product** belong in Git repos created to hold source to be published on <https://docs.microsoft.com>. Articles should include conceptual or procedural information required to understand and use the product. The goal of technical content is to show people **how** to do something, with a secondary focus on the **what** and the **why**.

Specific guidance exists for certain types of content, such as customer stories and service limits.

- **Case studies/customer stories:** In principle, customer stories/case studies can live either on the customer stories site (<https://customers.microsoft.com/>) or in the technical documentation repository, which publishes to <https://docs.microsoft.com/azure>. The difference is audience and technical depth. The customer-stories site addresses BDM and TDM audiences, and the content is not about technical deployment or operational details. The technical content channel addresses evaluators, implementers, and users and goes into technical depth appropriate for new to advanced users. It's crucial we follow procedure to ensure we are using partner companies' names with permission - see [the steps for publishing a case study or customer story](#).
- **Service limits:** Every service where subscription and service limits, quotas, and constraints apply must have an include file that documents the most important limits, quotas, or constraints. The include file must be added to the all-up "[Azure subscription and service limits, quotas, and constraints](#)" article. Optionally, you can publish a service-specific limits article to the service-specific content. The include file must be reused in that article.
- **Reference content:** Managed reference, REST APIs, PowerShell cmdlet help, schema reference, and error reference content is published to docs.microsoft.com. This content should be hosted separately in repositories that do not contain conceptual content. [Learn how to onboard reference content](#)

## Unauthorized content

Do not publish the following content on docs.microsoft.com or store them in an associated GitHub repository. These content types are delivered in other product or Microsoft content channels. Some types of content are not part of our content strategy.

- **Blog posts:** Blog posts are typically written in the first person voice and are related to announcements and promotions. They often sound like a personal story. This content typically belongs on the Azure blog. [Learn how to post a blog](#)
- **Code and project samples:** Code and project samples go in the samples repositories and are featured in the sample gallery. <https://docs.microsoft.com/samples/browse/?products=azure>

- **Community spotlight/community resources:** Do not publish articles featuring community projects. Docs.microsoft.com is for technical content about how to use the product or service described from the Microsoft perspective, not about how people are using the product. That's marketing or possibly blog content. Or, let the community tell its own story in the places that community likes best!
- **White papers:** White papers are typically .pdf files hosted outside of docs.microsoft.com. Links from docs.microsoft.com to white papers files can be made within article text, or in the table of contents, typically in a resources section. For more information, see [white papers](#).
- **Downloadable files:** Technical documents should be delivered as articles, not downloads. Rely on the OPS-provided "Download as PDF" feature of the publishing platform to make PDFs of the documentation available. Other downloadable content should go to the Microsoft Download Center. For more information, see [Where to host downloadable files](#).
- **Requests for customer feedback:**
  - Don't provide email addresses in technical articles for customer feedback. Docs.microsoft.com has many existing and monitored feedback methods. These methods include the feedback link that appears in the site footer, the satisfaction rating and verbatim control, the GitHub issues feedback, direct article contributions through GitHub pull requests, and product feedback sites such as UserVoice and Developer Community. Don't add to this plethora of existing channels by asking people to send feedback via email.
  - Articles asking for feedback are not technical content and should not be published. There are plenty of feedback methods available on the site.
  - If engineering/PM wants feedback on features in [public preview](#), they can create an Office form. Then, add this form link to your article. To create a form, see [Create a form with Microsoft Forms](#).
- **Pricing:** We don't publish pricing information for Azure on Docs. Pricing for Azure services is displayed on the pricing page on [azure.com](#) (ACOM). Here is background info on Azure pricing and the process for engineering teams:
  - **Why:** The ACOM pricing page can display pricing dynamically to allow for ongoing pricing changes and to show pricing in different currencies. Docs can't and shouldn't display pricing info, per agreement by Legal, ACOM, and the Docs teams. Finally, having the info in two places confuses users
  - **How:** Prices are sourced by ACOM from the Unified Pricing API (UPAPI), which in turn sources from the RateCard v2 API through the commerce catalog (Cayman)
  - **Problem:** ACOM does not hard code prices on the Azure pricing pages. Some services may not yet have meters configured in the commerce catalog
  - **Process:** Engineering teams should work with their PMM and business planning teams to create the meters. The PMM can make the request at the [Request Submission Tool](#).

If pricing meters cannot be created before the service launch date, and if the engineering teams want hard coding of prices on the ACOM pricing page, they need approval from these contacts:

Vivek Dalvi, Commerce Engineering  
 Richard Chin, Business Planning  
 Jeff Sandquist, DevRel  
 Cc: Srivatsan Kidambi, Business Planning and Isaac Hepworth, ACOM

**Note:** Any proposed change to this info needs to be reviewed by monicar (C+L) and sathota (ACOM)
- **Roadmaps:** We should avoid publishing product roadmaps on Docs for products that have a separate product update page. This applies particularly to Azure, although is a best practice for other products:

- **What:** Roadmaps convey high-level goals for a product. Azure previously published roadmaps but switched to the [Azure updates page](#), which is continually updated with service status in GA, preview, and development

- **Why:**

Marketing research shows that customers are satisfied with the Azure updates page. AWS and Google follow the same format

Roadmaps can be too speculative for technical content, they can be seen by competitors, they could negatively influence renewals, and they confuse users about where to find update info

- **Guidelines:**

Link to your product updates page when referring to recently released features. If there is no official MS page for your product updates you can optionally include a short list in your docs that is not speculative and that you continually refresh

If you have an existing roadmap article and your product has a separate product updates page, we recommend pointing to the official updates page, and discussing whether to phase out your article with stakeholders to avoid redundancy and confusion

There's more detail about the "why" in the next section (Future product plans or promises).

- **Future product plans or promises:** Do not publish statements about future product plans in technical documentation. Technical documentation should describe only what is possible in the released product. Why?
  - **It's not actionable:** Technical content should be about what customers can and should do today, not what they might be able to do some day.
  - **It erodes trust:** If a writer says a feature is coming soon and it is delayed or canceled, the statement risks eroding trust for all our technical documentation.
  - **Looks like marketing:** Talking about the future looks more like marketing material (aspirational rather than reality).
- **Marketing content:** Content that provides a high-level feature/benefit description or that just lists at a high level the capabilities of a service is probably marketing content. It belongs in marketing areas of the site. To publish marketing content, file a work request for [azure.microsoft.com](https://azure.microsoft.com).
- **Legal terms and software licenses:** Legal terms and software licensing are published to webpages such as [Microsoft Azure Legal Information](#) and [Microsoft License Terms](#). No general legal terms or software licensing should be published to [docs.microsoft.com](https://docs.microsoft.com). However, some product groups do publish more granular policies or terms to [docs.microsoft.com](https://docs.microsoft.com), under the guidance of their CELA representative. All groups must host this content in a centralized repo and follow a special localization process. For detailed information, see [Legal content](#).
- **Privacy information:** The all-up privacy policy for Microsoft Online Services covers all of Azure. Privacy information specific to a service should be presented as technical content, not "privacy statements". See <https://azure.microsoft.com/support/legal/>.
- **SLA:** Do not mention any specifics about SLAs in the technical documents. Always point to the SLA page for the service. The SLA index page is here: <https://azure.microsoft.com/support/legal/sla/>. Example: "For information about the SLA, see the [Azure service level agreements](#) page."
- **Open Source Software (OSS) technical documentation:** Technical documentation for Open Source Software should not be published to [docs.microsoft.com](https://docs.microsoft.com). The documentation files should be colocated in the same repository where the source code resides, making discovery easier. As with the source code itself, the community is also responsible for the review process and quality control of the documentation.

- **Pointer articles to downloads:** Instead of pointing small pages that contain nothing but a link to a download, just link to the download from the relevant technical content.
- **Private preview documentation:** The docs.microsoft.com site does not support private preview content. Product teams need to find other channels for documentation that supports products that are available only through private preview programs. Only technical documentation for publicly available services and software can be published to docs.microsoft.com. [Learn more](#).
- **Redirect articles:** Do not republish an article whose main purpose is to link to the replacement content. Convert the article into an actual redirect so the user doesn't have to click to go the replacement content. [Learn how to redirect](#).
- **Placeholders for future content:** Do not publish "coming soon" articles or sections as placeholders for future content. We only publish actual technical content articles that contain relevant, useful technical content.
- **Regions:** In general, avoid discussing or describing the regions in which an Azure feature, product, or service is available. Feature and service availability by region is provided to the public on the following ACOM page: <https://azure.microsoft.com/regions/services/>. In most cases, provide a link to the ACOM page.

However, the following exceptions are allowed:

- You can use specific regions in examples when you are describing procedures, tools, or developer endpoints. For example, this sentence is OK:

"The following screenshot shows two pings from two different region client machines, one in the East Asia region and one in the West US."

- You can specify region availability of service features if the ACOM page does not provide the level of granularity customers will need to successfully use the feature. For example, this sentence is OK if the feature is not covered on the ACOM page:

"Routing Preference support for storage account is available in the following Azure regions: France Central, North Central US, and West Central US."

#### **IMPORTANT**

For this exception, the article must contain `ms.custom: references_regions` in the metadata header for tracking purposes.

- **Release notes:** Release notes may be published only for products that are delivered as software or as an SDK. For example, in Azure, Storage Explorer, StorSimple, and SDKs are delivered as static, versioned products. Products that are delivered to customers as services are not normally versioned and therefore should not have release notes. The information you would have traditionally delivered upon release of new functionality should just be embedded in the relevant technical content or included in the [service updates channel](#) for the service.
- **What's new in a release or service:** Lists or descriptions of what is new in a release or a service go to the [Service Updates channel](#). For Azure Active Directory and Microsoft Intune, "What's New" articles are acceptable at this time because they are tenant-based, not subscription based.

## Non-Microsoft products

In Azure in particular, product and content efforts involve products and technologies not owned or created by Microsoft. Some recent examples:

- Microsoft enters into a partnership with another company and offers the product directly as an Azure service from the Azure portal.
- Microsoft acquires a technology that works with Azure and can drive adoption, but the technology maintains its separate branding (Cloudyn).
- We create content about how to use a widely used technology with Azure to reach communities that use that technology (Docker, Kubernetes).

We as Microsoft need to own our story. Content that directly relates to how to use anything with Azure or our other first-party products belongs on docs.microsoft.com. And, that content should be directly maintained by us. Content that is about how to use the partner product belongs on the partner's site to keep it distinct and simple. We can then link in our TOCs and content out to their site appropriately.

In these cases, the content channels would be:

- Overview of using the partner technology on Azure: docs.microsoft.com.
- Quickstarts and any Azure-specific tutorials: docs.microsoft.com.
- How-to content, feature-level content, and content about concepts specific to the partner technology: the partner's web site. We would link to this content from docs.microsoft.com.

# Respond to inappropriate content

5/19/2021 • 6 minutes to read

## NOTE



THIS DOCUMENT IS IN REVIEW AND IS NOT YET SUPPORTED IN THE CONTENT STANDARDS FOR [DOCS.MICROSOFT.COM](#). WE ENCOURAGE YOU TO USE THE GUIDANCE AND PROVIDE FEEDBACK [IN OUR TWO-QUESTION SURVEY](#).

As our content platforms continue to grow, we want to make sure all resources that we make accessible to the public are inclusive and appropriate. These guidelines apply to both content submitted from external users (such as in GitHub issues and comments) and content produced internally. This document provides guidelines for what to do when you see any content that might be considered inappropriate.

All of our content is governed by the [Microsoft Open Source Code of Conduct](#). This code lists the following as disrespectful and unacceptable behavior:

- Violent threats or language.
- Discriminatory or derogatory jokes and language.
- Posting sexually explicit or violent material.
- Posting, or threatening to post, people's personally identifying information ("doxing").
- Insults, especially those using discriminatory terms or slurs.
- Behavior that could be perceived as sexual attention.
- Advocating for or encouraging any of the above behaviors.

Trolling and spamming are also in violation of the [Microsoft Open Source Code of Conduct](#). For more information, see the [Code of Conduct FAQ](#) or contact [opencode@microsoft.com](mailto:opencode@microsoft.com) with questions or comments.

Additionally, we generally want to avoid publishing content with any words or phrases considered to be "sensitive terms" in our [PoliCheck database](#) (especially Sev 1 and Sev 2 terms).

There may also be content that doesn't fall into any of those categories, but which you still feel may be inappropriate. Even if you're not sure how such content should be handled, we encourage you to report it for further review. *If you see something – say something!*

## GitHub issues and comments

Occasionally, an external contributor may create a GitHub issue (or a comment left on an issue) that uses inappropriate language or otherwise violates the code of conduct. While this content is not authored by Microsoft, we take responsibility for ensuring that it is inclusive and appropriate.

If someone uses the feedback system in a way that appears to violate the Code of Conduct, email [opencode@microsoft.com](mailto:opencode@microsoft.com) with a link to the issue. The Open Code team will help you determine the appropriate action to take.

The resulting recommendation may include one or more of the following steps. You can loop in the C+L Repo Admins ([clrepoadmins@microsoft.com](mailto:clrepoadmins@microsoft.com)), who have permission to perform these actions in the repo.

- For a GitHub issue that is determined to have violated the code of contact:

- Locking the issue using GitHub's issue locking mechanism. The issue will no longer render on the docs.microsoft.com site, and community members will no longer be able to comment on the issue.
- Applying the `code-of-conduct` label to the issue.
- Closing the issue with a kind, but firm and professional comment, linking to the Code of Conduct. For example:

The [Microsoft Open Source Code of Conduct](#), which outlines the expectations for community interactions in and around docs.microsoft.com, is designed to help "provide a welcoming and inspiring community for all." The content of this issue appears to be out of sync with the code of conduct, so we have closed it.

- **For content that contains any explicit material or personally identifying information (for example, in the case of doxing):** Removing that content and adding a note to the top indicating "This post has been edited due to Code of Conduct violations".
- **For content where only one part is inappropriate, but the rest is OK:** Deleting the problem content with a message such as "Parts of this comment were removed as they violate the code [Microsoft Open Source Code of Conduct](#)".
- **For a user who has violated the Code of Conduct repeatedly:** Reporting the user to GitHub, or (as a last resort "nuclear option") [blocking them from collaborating in the repo](#).

You can also [report the issue or comment through GitHub's new content reporting mechanism](#), in addition to emailing opencode@microsoft.com. However, it can take a long time for the report to be reviewed using this method.

## Archived blog content on docs

Many years ago, Microsoft hosted a blog platform for employees to use in whatever way they wanted. While this platform is no longer available, most of its content has been archived at

<https://docs.microsoft.com/archive/blogs/>. As part of this process, blog posts that included Sev 1 and Sev 2 Polcheck terms were removed, along with some other pages that were not appropriate for other reasons. While we believe that these efforts removed most of the problematic content, it's possible that you might still come across a page that should be removed.

To report inappropriate content in the blog archives:

1. Go to <https://sitehelp.microsoft.com>.
2. Click the **Submit request or issues** button in the header.
3. In the **Select Service** drop-down list, select **Archived MSDN/TechNet Blogs**.
4. In the **Select Service Category** drop-down list, select **Report offensive/inappropriate content**.
5. Enter the required information, then select **Submit**.

The SRE (Site Reliability Engineering) team will create a ticket for the issue within 24 hours. The PM for this area will triage the request and work with the engineering team to remove the topic if needed. The ETA to remove may depend on the type of request, and the work required. Once the issue is resolved, SRE will close the request and the requestor will be notified by the Devops ticket system.

## Docs content

If you find anything inappropriate in a page on docs.microsoft.com (excluding blog archives), you can notify the page owner or team directly so that they can take action.

Alternately, you can create a GitHub issue by using the feedback mechanism at the bottom of the page.

To create an issue:

1. Scroll to the bottom of the page. In the **Feedback** section, select **This page**.
2. Add a title such as "Code of Conduct violation."
3. Replace **[Enter feedback here]** with a brief description of the issue. Keep in mind that external customers may see what you add here.
4. Select **Submit new issue**.

**NOTE**

If the docs page doesn't have a **Feedback** section, and you don't know who owns it, you can file a SiteHelp issue as described in the **Archived blog content on docs** section.

## Learn content

For inappropriate content on docs.microsoft.com/learn, you can notify the content owner or team directly so that they can take action.

Alternately, you can report an issue via the feedback mechanism at the bottom of the page.

To report an issue:

1. Scroll to the bottom of the page. In the **Need help?** section, select **reporting an issue**.
2. Select **Learning content quality**, then select **Other**.
3. Enter a title such as "Code of Conduct violation."
4. Provide a brief description of the issue. You may want to include your alias in case anything needs to be discussed.
5. Select **Submit**.

## Responding to reported violations in your content

If you are the owner of a page that someone reports for having inappropriate content, please address the issue as soon as possible by making the appropriate edits to remove or update the content.

If you need help determining how to handle a report, or have any questions, contact [opencode@microsoft.com](mailto:opencode@microsoft.com).

## For more information

The [Microsoft Open Source Code of Conduct Enforcement Course](#) helps you learn more about the code of conduct, how to recognize violations, and how the resolution process works.

**NOTE**

If you are working on a product that is open-sourced, verify the code of conduct that applies to your repo. For example, repos in the dotnet GitHub organization are covered by the [.NET Foundation Code of Conduct](#). The principles and the text are almost exactly the same as the Microsoft Open Source Code of Conduct, but the addresses for reporting violations are different.

# Customer stories and case studies

4/27/2021 • 2 minutes to read

In principle, customer stories/case studies can live either on the [Customer Stories](#) site or in the relevant technical documentation repository, which publishes to [docs.microsoft.com](#). The difference is audience and technical depth. The [Customer Stories](#) site addresses Business Development Manager and Technical Delivery Manager audiences, and the content isn't about technical deployment or operational details. The technical content channel addresses evaluators, implementers, and users and goes into technical depth appropriate for new to advanced users.

Process details:

- When a technical content author submits a customer story or case study through the technical content channel, they must contact the [Cloud Marketing Advocacy & Evidence team](#) to request a review of the content. CC M. Baldwin ([mbaldwin@microsoft.com](mailto:mbaldwin@microsoft.com)) while you're at it. The Advocacy & Evidence team will evaluate the content.
  - If the Advocacy & Evidence team feels the content is appropriate to publish on the [Customer Stories](#) site, they'll reply with instructions on next steps. The author will work with the team to get the content published.
  - The Advocacy & Evidence team may feel that the content's technical depth means the content is better suited for publication through the technical content pipeline. If that happens, the team will continue with its standard quality review process and publish when ready.
  - In either case, to publish the content, the content author must present an email from a representative of the partner company. The email should indicate that:
    - The partner company agrees to serve as a reference.
    - We have permission to use their company name in the customer story or case study.If a representative of that company later contacts Microsoft to remove the content, we'll remove it immediately.
- If the content author submits a highly technical case study or customer story through the customer stories channel, the Advocacy & Evidence team can choose to contact the Technical Content team to request that they publish the content through the technical content channel.
- In both cases, content must meet the quality criteria and content guidance for the respective channel.
- If the content author develops technical content to support a customer story on the [Customer Stories](#) site, the technical content author should contact the Advocacy & Evidence team to add cross-links to the supporting technical content.
- After the team publishes the customer story or case study to the [Customer Stories](#) site or [docs.microsoft.com](#), the author or PM must submit the details on the solution, products, and customer overview via [the nomination form](#). The Advocacy & Evidence team will vet the details with the product teams to assess the lead against evidence gaps and alignment to Global Engagement Programs. The team may require follow-up work as a result of this process.

# White papers

5/20/2021 • 2 minutes to read

This article describes what white papers are and how to deliver them. We'll update these guidelines as we consider new ways to use white papers and other long-form content more strategically.

## What is a white paper?

A white paper is long-form content that communicates strategy, methodology, new technologies, or a technical or marketing scenario. For example: [Architecting for the Cloud \(AWS\)](#), [Google security white paper](#), and [Azure monitoring and analytics at scale](#).

## How to deliver

- **File format:** We recommend using a PDF file format for white papers. To create a PDF within Microsoft Word, select **File > Export**, and then choose **Create PDF/XPS Document**.
- **Share on Docs:** The [White papers for Power BI article](#) is a good example of how to share white papers within documentation. This article lists white paper links and provides an abstract of each one. White paper files are hosted outside of the Microsoft docs site, either on [azure.com \(ACOM\)](#) or the [Download Center](#).
- **Host on ACOM:** Azure has a site dedicated to sorting and searching white paper content: [White papers, analyst reports, and e-books](#). To upload a white paper to the Azure site, file a request at the [ACOM portal](#). **Contacts:** Sita Dontharaju and Lisa Tofmark.
- **Host on Download Center:** You can host a .pdf file on the Microsoft Download Center by using the [Download Management Studio](#). This tool offers both a download URL and a landing page on the Microsoft Download Center to host the URL. If you have the option of hosting your white paper on Microsoft docs or ACOM, create only the URL, not the landing page.
- **Create short URLs:** Use aka.ms to create shortened links with a vanity URL to point directly to a white paper PDF download. These links come in handy for social media sharing and marketing purposes. They also have built-in analytics to count the number of hits. Example vanity URL: <https://aka.ms/powerbi-b2b-whitepaper>. Use the [Aka Link Manager](#) to create a shortened URL.

## See also

- [White papers, analyst reports, and e-books](#)
- [AWS white papers](#)
- [Google Cloud white papers](#)

# Blog process and guidelines

3/5/2021 • 3 minutes to read

Here's how to submit a blog post and how to distinguish between a [TechCommunity blog post](#) and technical documentation. There's also an example blog post that points to the technical article with more detail.

## Submit a blog post

- **Start a blog post:** [Use the C + AI blog post staging request form](#). Once you submit the form you'll be contacted by the C + AI blog team. It takes about four days to complete the review and staging process.
- **Publish a blog post:** Same form as above.
- **Fix a mistake:** Contact the C + AI Blog Team alias with the blog post URL and the fixes you want to make. The team will make the edits and can usually do so in the same business day.
- **Current blog scope:**
  - Azure
  - Open source
  - Industry
  - Visual Studio
  - SQL Server
  - Machine Learning
  - IoT
  - Power BI
  - Dynamics 365
  - Quantum

## Blog vs. documentation

### Blog

Blog posts on the tech community pages should introduce the topic and explain why it's notable or useful. They're our public face to show thought leadership and expert input on use cases for the feature. Some key distinguishing features of a blog post:

- Typically explains "**the what and the why**": What is this new feature or service? Why will I, the reader, want to use it?
- Is intended to be persuasive and tell users about a product's benefits.
- Can refer to future updates.
- Background information and timely info for here and now.
- Feature announcements with links to docs for how-to.
- **Supplemental** info to the docs.
- Occasionally, for edge cases, includes how-to content.

### Documentation

Content on Microsoft's Docs platform is tutorial-focused and intended to provide evergreen how-to guidance. Some key distinguishing features of docs content:

- Typically explains "**the how**": How can I use this feature or service? Detailed step-by-step guidance for the

standard scenarios.

- Is factual and helps users understand and use the product.
- Is updated frequently to reflect the current state of the product.
- Doesn't refer to future updates. Technical docs convey how something works, not how it might work.
- Is written consistently across all MS services for standardization of voice, look and feel, and behavior.

## Blog and article examples

Here's a blog entry that explains and points users to the corresponding article for details, updates, and the larger context.

### **Overview: memory grant feedback in batch mode**

SQL Server uses memory to store in-transit rows for hash join and sort operations. When a query execution plan is compiled for a statement, SQL Server estimates both the minimum required memory needed for execution and the ideal memory grant size needed to have all rows in memory...

If the cardinality estimates are inaccurate, performance can suffer:

- For cardinality underestimates ...
- For cardinality overestimates ...

Adaptive memory grant feedback in batch mode in SQL Server is a way to correct the grant value through repeated executions of a query.

This feature is the first improvement under the adaptive query processing family of features to be surfaced in the public preview of the next release of SQL Server on Linux and Windows ...

Read the article: [Adaptive query processing in SQL databases](#) describes the feature in more detail. Updates to the feature will be made to the article, which answers these questions and links you to how-to and other info:

- How does adaptive memory grant feedback in batch mode improve memory grant sizing?
- What kind of results can I expect?
- Can I see the adjusted memory grant in my execution plan?
- How do I enable adaptive memory grant feedback in batch mode?
- What if my memory grant requirements differ significantly based on parameter values of consecutive executions?

### **Keep in touch**

If you have any questions about any connectivity for Azure SQL Database, please contact [someone@microsoft.com](mailto:someone@microsoft.com).

## See also

- [Quick tips for creating a blog post](#)

# Where to host downloadable files

4/16/2021 • 2 minutes to read

Sometimes, teams need to post content in the form of a Word document, a compressed file, PDF, or other "downloadable" file format. Technical content repositories on GitHub aren't intended as hosting locations for these types of content. Content in the MicrosoftDocs repositories should be limited to technical documents authored as markdown or YML files and images used in those documents.

The Microsoft Download Center is the correct channel for posting downloadable content.

For more information, see the [Microsoft Download Center \(DLC\) documentation](#).

For more information about what content goes where, see [the content channel guidance](#).

# SEO onboarding for contributors

5/21/2021 • 2 minutes to read

This article provides resources to help you incorporate basic SEO into your daily writing practices.

Common questions about SEO for tech docs are "Where do I start?" and "Will SEO take lots of time?" The approach to SEO for Microsoft Docs takes into account unique aspects of the web as a medium and best practices for writing for it. The guidance is a matter of tuning up your writing practices, not adding another step to authoring.

## Introduction and fun facts

Google Search is a primary driver of content discovery on the web. 90% of MS Docs organic search traffic is from Google Search, which dominates the search market. Therefore, SEO practices here are based on Google Search and its data.

Before you dive into SEO, consider how organic search relates to docs:

- **Google Search** is a content ecosystem. It includes clickable results and answers on the search page you don't need to click.
- **Search data is customer data.** It provides data on the terms they use, subjects they are interested in, questions they ask, and the search results they click.
- **Writing well for search is a matter of writing well for the web.** Google Search uses semantic HTML structure to understand what content is about and how to serve up results. Understanding the medium of the web and your audience is key to writing search-friendly content.

## Beginner resources

### Basic writing practices

- [Guidance for the SEO cheat sheet](#)
- [SEO cheat sheet](#): The basic techniques condensed to one page.

### Data sources

- [Microsoft Docs Metrics](#): Data for a single doc page.

### Tools & how they help

Learn about how your content performs in search and what your customers are searching for.

- [Google Search](#): Use it daily for basic research into your technology domain and terminology. Check the search experience for your docs. Search on common, unbranded phrases describing doc intents to see where your competitors' content ranks and on what terms.
- [BrightEdge](#): Keyword research and URL analytics using the DataCube. Email [Carolyn Gronlund](#) for an account.

## Intermediate resources

### Tune up a doc page

[Checklist for web-friendly docs](#): Guidance on using doc page elements effectively.

### Data sources

- [Google Search Console](#): Data on the top 1,000 pages and KWs for the site or part of site specified. Also, info

about crawl and indexing errors. Get access from [Khairun Jamal](#).

- [Content Performance dashboard for docs](#) (C&L): Shows referrers including organic search and provide links to analyze referrers in Kusto. About 51% of website traffic on the internet is from organic search, according to BrightEdge. Each doc might not achieve this rate of search referrals, but it's a good goal for your doc set and for strategically important content.
- [Low/No PV Dashboard](#): Unpublishing low- or zero-pageview content is important to site health and the Google quality metric that influences search rank. Our other dashboards don't show content that gets zero pageviews.

### Tools & how they help

Dig deeper into KW research and to track your choices.

- [SEMRush](#): Keyword research and URL analysis. Shared accounts assigned to various team members.
- [STAT Analytics](#): Keyword tracking data and trends. Only provides data on the KWs you enter. Get read access to the KW universe and write access to your own project.

# Guidance for using the SEO cheat sheet

5/21/2021 • 6 minutes to read

This article covers simple techniques that help content perform well in search, the primary driver of traffic to Microsoft Docs. These practices are core to a web writer's skillset: *Make them habits*.

Use this guidance to help you incorporate the [SEO cheat sheet for writers](#) into your writing.

## Why does search matter?

Search is how most customers find information on the web. Some data points to consider:

- Google search is the single highest-volume source of traffic for Content & Learning docs, accounting for 90% of organic search traffic.
- 75% of search page clicks are on the *top three search results on page 1* of Google, according to a study by [Backlinko](#).
- Search is a zero-sum game: If your docs don't rank well, competing content will.

## Be found in search: Page title, description, and friendly URL

Your page title has the greatest impact on search rank and traffic. Along with the meta description and friendly filename/URL, it's one of the three elements of a standard search result listing.

The words in a search result help readers determine its relevance. The article page title and friendly filename directly impact search rank.



A *search result with a page title, URL, and meta description. The publishing system adds the site identifier "Microsoft Docs"*.

---	
title: Copy data from Blob Storage to SQL Database - Azure SQL Database	
description: This tutorial shows you how to use Copy Activity in an Azure Data Factory pipeline to copy data from Blob storage to SQL Database.	
keywords: Google doesn't use keywords to determine rank.	

TAG	GUIDANCE	NOTES
title (length)	Character counts are estimates: <a href="#">Check a preview</a> . Title body is about 60-65 chars max including spaces and brand.	Unique text string with important terms first. The article intent must be clear in search results.

TAG	GUIDANCE	NOTES
title (branding)	Include the product or technology, such as Azure or ASPNET.	Add brand using the titleSuffix tag in the metadata or in the docfx.json file in the doc repo. If brand is essential to the title intent, ensure it displays in the search result
description	Minimum 100, max 160 characters including spaces.	Displays on the search page inline with the article date stamp. If your intro para describes your article's intent, you can use it here edited for length.
keywords	<b>Don't use this tag</b>	Google doesn't use this tag for search indexing and rank.
Friendly URL / filename	80 chars or fewer. Only lowercase letters, numbers, and dashes.	Follow filename/friendly URL guidance for your documentation set.

## Establish relevance quickly: On-page text

The Heading 1 (H1, not called a *title*) has the second greatest impact on search traffic and rank.

In just a few seconds, a reader determines what a page is about and whether it's relevant. The H1 and first paragraph (intro paragraph) are key to establishing relevance quickly.

# What are the different Hadoop components available with HDInsight?

12/5/2016 • 6 min to read • Contributors  all

Find out about the different service levels offered by HDInsight as well as the versions of different Hadoop components included with HDInsight.

**Relevance:** How well a search result or webpage quickly answers the questions "Is this the content I wanted?" or "What's the intent of this page?"

```
# What are the Hadoop components available with HDInsight?
```

Find out which Hadoop components and versions are included with HDInsight, as well as the different service levels available.

TEXT	GUIDANCE	NOTES
H1	100 chars or fewer, including spaces.	Must be unique, with most important or differentiating terms first. You can repeat the page title, but you also have room to elaborate.
1st paragraph	The primary intent of the article in 1 or 2 short sentences.	The first para can be edited for reuse as a meta description.

## Make an article scannable: H2s

The heading 2 (H2) is a key element that makes articles scannable. The H2 is a significant contributor to search rank and on-page relevance.

Google displays H2s as:

- Links in a search result, when relevant to the query.
- The title of the result, if highly relevant to the query.

Principles for H2s:

- **Include enough detail to describe the section intent:** What will a reader learn or accomplish?
- **Pay attention to on-page display:** The H2 should display without excessive wrapping under "In this article".
- **Avoid one-word H2s:** Exceptions are standard headings specified for content types, such as "Prerequisites" for tutorials. Otherwise, a single word is rarely adequate to describe an intent.

## Improve search rank with diagrams: Image alt text and filename

Well-written image text - alt text and a friendly image filename - can help improve an article's overall search rank.

Alt text describes the image for people who use assistive devices. You can also use alt text re-emphasize the article intent and keywords. Alt text and the friendly filename should use the same key terms.



*Image results on page 1 for "dashboard" from Microsoft Docs content.*

### TIP

For SEO performance, prioritize diagrams over screenshots when optimizing alt text and image filenames. A diagram is likely to rank in search if its alt text and filename address a customer intent. **All images require alt text for accessibility.**

![A dashboard tile is a snapshot of data, pinned to a Power BI dashboard.](.media/service-dashboard-tiles/power-bi-dashboard.png)

TEXT	GUIDANCE	NOTES
Image alt text	Min. 40 chars; max 150 chars including spaces. Put the most descriptive keywords first.	Descriptive alt text coupled with a good image filename can help your content rank on search page 1 in the image carousel.

TEXT	GUIDANCE	NOTES
Image filename	80 chars or fewer, including dashes. Only lowercase letters, numerals, and dashes.	Write a descriptive friendly filename, much like a friendly URL. It is fine to use a brand, product or technology name in the image filename. A good image filename and descriptive alt text can help your content rank on search page 1 in the image carousel.

## Use common terms people search on

You can write great content, but if it doesn't contain words and phrases people use in search, nobody will find it.

### Basic terminology research tools

How do I know what words people search on? A few ideas:

- **Search on phrases describing your primary intents.**
  - Try synonyms and variations on phrases.
  - Look at search results. If you see relevant results and competitors' content, you're on the right track.
  - Explore competitor content and see what terms they use.
  - Talk to customers. How do they describe what they are trying to do?
  - Check sources such as StackOverflow, Reddit, and so on.
- **Check search suggestions in Google.**
  - Type a term and see what phrases Google suggests. Also, check related searches.
- **Answer the Public:** Questions and phrases used in search on Google and Bing.
  - Be sure to choose US from the region selector.
  - Enter a term of 1-2 words to see search phrases that use it.

### Unbranded terms help customers find your article

Wherever possible, use [unbranded terms](#) to describe the article intent. Branded terms are Microsoft brands, product/service names, and feature names. Unbranded terms are open-source or other standard terms used commonly in the industry, but not owned by a company.

- **Don't write this H1:** Create an Azure SQL Database elastic database pool

This heading relies solely on branded service and feature names. It assumes readers know what an elastic pool is used for (big mistake!), rather than describing a scenario in terms customers commonly use.

- **Do write this H1:** Manage multiple databases with an elastic database pool in SQL Database

At the front of this heading is unbranded terminology describing the customer scenario/intent: managing multiple databases.

## SEO hackathon: Improve search performance for a content set

A great way to make improvements in search discovery quickly is to conduct an SEO hackathon. Using SEO bug reports and the [SEO cheat sheet for writers](#), you can quickly set up a hackathon session and measure results for your work.

See [How to run an SEO hackathon](#).

# SEO: Checklist for web-friendly docs

6/2/2021 • 3 minutes to read

This guidance is a checklist for writing web-friendly doc pages. The list isn't a substitute for an edit pass, but it can help you quickly find issues with a doc page that may be hurting performance.

By following this checklist, you can improve the search performance, engagement, readability, and relevance of a doc article.

## The checklist

- [Is the page intent clear?](#)
- [Is there only one intent?](#)
- [Does the intent use common terminology?](#)
- [Does it follow the SEO cheat sheet?](#)
- [Is the product/technology brand in the page title?](#)
- [Does the article have relevant cross-links?](#)
- [Is the article scannable?](#)
- [Are tables short and scannable?](#)
- [Do diagrams use text sparingly?](#)

## Write a single, unique intent

### **Is the page intent clear?**

Find a succinct, customer-friendly way to state the intent, and then repeat it in the:

- page title
- meta description
- H1
- intro (first) paragraph

The intent should be unique to this article; don't repeat H1s or page titles in other articles.

**Action:** Write a clear page intent, and repeat it in the title, meta description, H1, and intro para using the same customer-friendly terms.

Page intent repeated in title, H1, and intro para:

The screenshot shows a Microsoft Edge browser window displaying the Azure Machine Learning documentation. The title of the page is 'Create a data labeling project and export labels'. A red box highlights the title and the main content area. On the left, there's a sidebar with navigation links like 'Get data labeled', 'Label images', etc. On the right, there's a sidebar with 'In this article' sections and a 'Feedback' button.

## Is there only one intent per page?

This is a usability and engagement issue, as well as an SEO issue. If your page has multiple intents, it might bury an important user intent, be difficult to follow, or simply be too long.

**Action:** Consider splitting up articles that have multiple intents.

## Does the intent use customer and industry terminology?

Use common industry terminology to describe the intent and don't depend on Microsoft proprietary feature names or product brands.

Identifying industry terms is as simple as spending 15 minutes checking Google Search, competitor docs, open-source docs, industry publications, and blogs. Researching the industry standard and competitive terminology is part of acquiring domain knowledge in your technical area.

### Actions:

- Describe the scenario/intent using common industry terms.
- Find common ways to describe scenarios and functionality using Google Search and other web sources.
- Define each acronym you use, unless it's common, such as *XML*.

## SEO basics

### Does it follow the SEO cheat sheet?

The [SEO cheat sheet](#) isn't just about search-optimizing a doc page; it's about writing with the user experience in mind.

**Action:** Incorporate the SEO cheat sheet guidance into your everyday web writing habits.

### Is the product or technology brand in each title?

Brand or technology names provide a key relevance signal for search results and contribute to rank. Longer brands might truncate on the search page, and that's OK.

**Action:** Use the titleSuffix property in the article metadata or docfx.json file.

### Link to and from highly relevant docs

Cross-links help the reader find related content and help Google understand relationships in content. Relevant cross-links to and from an article can raise its search rank.

**Action:** Add links to and from closely related content. If another doc has closely-related intent, consider adding cross-links between docs just below the intro paras. Don't make the customer scroll far to find an important link.

# Effective formatting

## Is the article scannable?

Use informative headings and page organization that help customers skim the article and quickly find information.

Readers don't track hierarchical relationships in headings deeper than three levels, so keep headings shallow. It's not necessary to maintain an exact logical hierarchy, especially if it leads to more than three levels of headings. Avoid H4s, and don't ever use deeper heading hierarchies.

### Actions:

- Use shallow heading hierarchies: H1, H2, and H3 only.
- Write informative subheadings that help readers find information quickly.

## Are tables scannable?

Avoid using long tables that require scrolling. They aren't scannable and can hide key information from customers and search crawlers.

Action: Break up long tables thematically with H2s.

Example of a long table broken up by H2s from Azure Architecture content: [Azure to AWS comparison](#).

## Data orchestration / ETL

AWS service	Azure service	Description
Data Pipeline ↗, Glue ↗	Data Factory ↗	Processes and moves data between different compute and storage services, as well as on-premises data sources at specified intervals. Create, schedule, orchestrate, and manage data pipelines.
Glue ↗	Data Catalog ↗	A fully managed service that serves as a system of registration and system of discovery for enterprise data sources
Dynamo DB ↗	Table Storage ↗, Cosmos DB ↗	NoSQL key-value store for rapid development using massive semi-structured datasets.

## Analytics and visualization

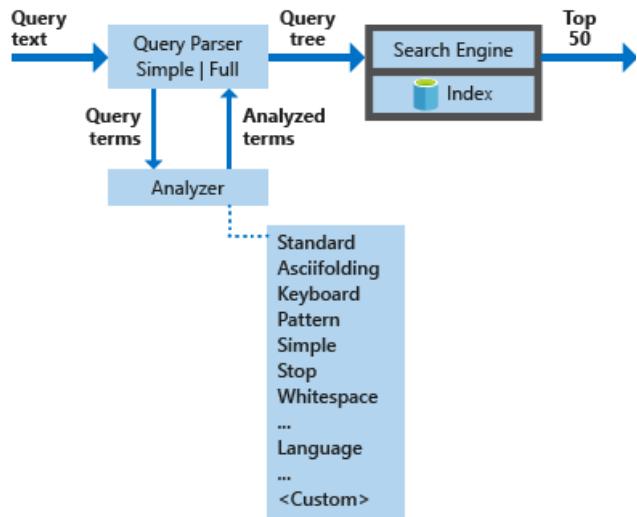
AWS service	Azure service	Description
Kinesis Analytics ↗	Stream Analytics ↗	Storage and analysis platforms that create insights from large quantities of data, or data that originates from many sources.
	Azure Data Explorer ↗	
	Data Lake Analytics ↗	
	Data Lake Store ↗	
QuickSight ↗	Power BI ↗	Business intelligence tools that build visualizations, perform ad hoc analysis, and develop business insights from data.

## Do diagrams use text sparingly?

Avoid text-heavy diagrams. Text in diagrams isn't accessible, and Google can't crawl it.

**Action:** For any diagrams with a significant amount of text, incorporate the text into the article body text instead.

Diagram with text that is reinforced in the article to mitigate discoverability or accessibility issues:



A text-heavy diagram with information that isn't found in accessible text:

## Inbound link considerations

An inbound link to a webpage is a link from a website, document, or client. The relevancy of link context and link text to the target page impact how the page performs.

Be aware of where there are links your docs, if this data is available.

Are there inbound links to the page from:

- A product portal or UI?
  - Marketing pages?
  - Microsoft blogs or other corporate content?

**Action:** Align the content of your page with internal cross links, or work with partners to change the cross links.

# SEO: Tips for writing titles

11/2/2020 • 6 minutes to read

This article helps you write better titles, by using examples from published documentation. The titles follow guidance from [SEO basics](#), a useful guide to basic SEO techniques.

The title tag is important for SEO and visitors. The title string displays as the content title in the SERP (search engine result page) and in browser tabs. It's the most important factor in search ranking of the article. Its primary job is to tell visitors and search engines what they can expect from the web page in the most concise way possible.

## IMPORTANT

Don't confuse title with H1; they are two different text elements. The H1 displays at the top of the doc page and has a limit of 100 characters including spaces. The title tag is in article metadata and displays in the browser bar and on the search engine result page.

## Rules for writing titles

- **Preview the title text for clear intent.** Paste the title into [Moz title tag preview](#).
  - Is there enough information to understand what the article is about?
  - Are key, differentiating terms and keywords at the head of the title?
- **Optimal: Maximum 60-65 chars including spaces and brand (titleSuffix), but *preview* your title.** Don't sacrifice clarity or service/product brand to fit.
- **Product brand is required.** Brand can truncate in the search result as long as the article intent is clear. For Azure services, make sure the service brand is in the title. It can be appended using spaces and a hyphen character:

Description of article content - Product Brand

- Words that truncate in search results still help with search rank, but don't help users determine relevance.
- titleSuffix metadata can be used to enter brand on each article and at folder level and globally in docfx.json file.

## Check your title and description at author time in VS Code

The [Docs Authoring Pack for VS Code](#) now includes a Search Results Preview to help verify that your title and description will be helpful for users when returned in search. Using publicly available information from Google, Search Results Preview generates an approximation of what your article title and description will look like when returned in Google search on Chrome with default font settings. This is the most common way people find content on Docs, accounting for more than half of all page hits. Because much of Google's algorithm is secret, the preview might not match exactly. But it gives you an idea of how your title will be truncated and what description users will see when they find your article via search. Use it to make sure you're providing the most helpful, relevant information.

To use Search Results Preview:

1. In VS Code with the Docs Authoring Pack, click `Alt+M` to access the Docs Markdown menu or `F1` to access

the command palette.

2. Select **Search Results Preview** from the Docs Markdown menu or filter the command palette to find **Docs: Search Results Preview**.
3. Preview will open in a side-by-side window.

Here's search preview for the current article:

docs.microsoft.com ▾

## Docs content discoverability in Search - SEO - Docs Contributor...

Nov 2, 2018 - Learn how to setup SEO Runs for metadata issues and troubleshoot SEO issues.

## Meta title examples

**Example 1:** <https://docs.microsoft.com/azure/virtual-machines/linux/tutorial-manage-vm>

**Current title tag:** Tutorial - Create and manage Linux VMs with the Azure CLI (57 characters)

**Edited title tag :** Tutorial: Create & manage Linux VMs with Azure CLI (50 characters)

### TIP

To reduce title tag length, try removing connecting words and articles like "and" and "the". A great way to replace "and" is with an ampersand "&", and often "the" can be omitted without hurting the meaning of the title tag. For tutorials, you use "Tutorial" in the title to call attention to them and help them rank in search.

**Example 2:** <https://docs.microsoft.com/azure/security-center/security-center-alerts-data-services>

**Current title tag:** Threat detection for data services in Azure Security Center (59 characters)

**Edited title tag:** Data services threat detection – Azure Security Center (54 characters)

### TIP

This example highlights the value of rearranging a phrase in order to reduce character length. In this case, switching the order of "Threat detection for data services" to "Data services threat detection" eliminated the need for the word "for" and saved some characters. Replacing "in" with a dash also minimized the length by a couple of characters.

**Example 3:** <https://docs.microsoft.com/azure/security-center/security-center-just-in-time>

**Current title tag:** Just-in-time virtual machine access in Azure Security Center (60 characters)

**Edited title tag:** Just-in-time VM access – Azure Security Center (46 characters)

### TIP

Another great way to shorten title tags is by replacing industry terms with common acronyms, when they are widely known and adopted. In this example, using the acronym for "virtual machine", "VM", shortened the title tag by quite a bit. Replacing "in" with a dash also reduced the length by a couple of characters. The meaning of the title tag remains the same, but it is now shorter.

**Example 4:** <https://docs.microsoft.com/sql relational-databases/logs/troubleshoot-a-full-transaction-log-sql->

## server-error-9002

Current title tag: Troubleshoot a Full Transaction Log (SQL Server Error 9002) - SQL Server (72 characters)

Edited title tag: Error 9002: Troubleshoot transaction log – SQL Server (53 characters)

### TIP

With this example, the title tag was shortened by removing some repetitive phrasing (SQL Server mentioned a second time). The error code was moved to the head of the title, because keyword research shows users often search on them. A few optional words ("full", "a") were also removed to reduce the overall length. The SQL Server brand is programmatically appended in both versions.

Example 5: <https://docs.microsoft.com/learn/modules/build-ml-model-with-azure-stream-analytics/>

Current title tag: Analyze images in real time with machine learning, Azure IoT Hub, and Azure Stream Analytics - Learn (100 characters)

Edited title tag:: Analyze images with machine learning, Azure IoT hub, & Stream Analytics - Learn (79 characters)

### TIP

In this example shortened the length by removing the extra words and keeping all the keywords intact including the brand.

Example 6: <https://docs.microsoft.com/aspnet/core/host-and-deploy/azure-iis-errors-reference>

Current title tag: Common errors reference for Azure App Service and IIS with ASP.NET Core (71 characters)

Edited title tag: Common errors - Azure App Service & IIS with ASP.NET Core (57 characters).

### TIP

With this example, a very specific title tag was shortened by removing some optional words and adding a dash. These tweaks tightened the wording of the title tag so the entire title intent is readable on the search page.

## How to add brand using titleSuffix metadata

Title suffix metadata allows you to enter brand using different methods. As any other metadata topic level metadata overwrites folder level and global metadata. Here are the examples for each type.

**Topic Level** You can add titleSuffix metadata on each article to append the title body with brand.

```
---
title: Model training methods
titleSuffix: Azure Machine Learning
description: Learn the different methods you can use to train model with Azure Machine Learning. Estimators provide an easy way to work with popular frameworks like Scikit-learn, TensorFlow, Keras, PyTorch, and Chainer. Machine Learning pipelines make it easy to schedule unattended runs, use heterogenous compute environments, and reuse parts of your workflow. And run configurations provide granular control over the computation targets that the training process runs on.
services: machine-learning
ms.service: machine-learning
author: Blackmist
ms.author: larryfr
ms.subservice: core
ms.topic: conceptual
ms.date: 09/18/2019
---
```

### IMPORTANT

Make sure titleSuffix metadata is entered with exact casing as "titleSuffix" with uppercase "S"

In the previous example, the full title in search becomes "Model training methods - Azure Machine Learning | Microsoft Docs". The branding in titleSuffix contributes to the overall title length, so avoid repeating the full brand name in the title metadata. The advantage of this method is that all articles are consistently branded for search ranking.

**Folder Level (docfx.json)** Folder level metadata is entered in docfx.json file. Here is an example from Azure-docs-pr repo

```
"titleSuffix": {  
    "articles/governance/blueprints/**/*.md": "Azure Blueprints",  
    "articles/governance/blueprints/**/*.yml": "Azure Blueprints",  
    "articles/governance/policy/**/*.md": "Azure Policy",  
    "articles/governance/policy/**/*.yml": "Azure Policy",  
    "articles/governance/resource-graph/**/*.md": "Azure Resource Graph",  
    "articles/governance/resource-graph/**/*.yml": "Azure Resource Graph",  
    "articles/lighthouse/**/*.md": "Azure Lighthouse",  
    "articles/industry/agriculture/**/*.md": "Azure for Industry: Agriculture",  
    "articles/sql-data-warehouse/**/*.md": "Azure SQL Data Warehouse",  
    "articles/sql-data-warehouse/**/*.yml": "Azure SQL Data Warehouse",  
    "articles/data-factory/**/*.md": "Azure Data Factory",  
    "articles/data-factory/**/*.yml": "Azure Data Factory",  
    "articles/sql-database/**/*.md": "Azure SQL Database",  
    "articles/sql-database/**/*.yml": "Azure SQL Database"
```

**Global metadata (docfx.json)** Brand (titleSuffix) can be set globally under Global metadata section of docfx.json file.

Here is an example from Xamarin repo

```
2     "globalMetadata": {  
3         "uhfHeaderId": "MSDocsHeader-Xamarin",  
4         "breadcrumb_path": "/xamarin/breadcrumb/toc.json",  
5         "titleSuffix": "Xamarin",   
6         "extendBreadcrumb": true,
```

#### NOTE

For large repos with docs for many services and technologies, make sure you apply titleSuffix to the correct folder level titleSuffix. Setting titleSuffix for an entire repo is useful only for repos containing docs for a product or service with a single brand.

# SEO: How to write good meta descriptions

9/12/2019 • 3 minutes to read

The meta description is an article summary that displays under page titles on the search engine results page (SERP) as well as in the Docs site search feature. It describes the benefit or intent of the article. Even though it's not crawled by search engines for rank, it helps with click-through rate.

## Rules for meta descriptions

- Maximum of ~160 characters and minimum of ~100 characters including spaces. (Google reverted from its recent experiment with longer meta descriptions.)
- At least one full sentence that describes the page topic fully.
- Use text from the introductory paragraph of on-page copy edited for length, if it describes the article intent.
- Write key concepts at the head of the description.
- Optional: Include a CTA (call to action) in the meta description. Example from H&R Block website (CTA in *italics*): "Not sure if your dependent children are required to file taxes? *Learn more about tax rules for children with the experts at H&R Block.*"

## Meta description examples

**Example 1:** <https://docs.microsoft.com/learn/patterns/get-started-with-marketing/>

**Current meta description:** Microsoft Dynamics 365 for Marketing is a marketing automation application that enables you to turn prospects into business relationships. You can find and nurture more sales-ready leads by moving beyond basic email marketing. Connect sales and marketing, automate processes, and make smarter decisions to maximize your marketing ROI. Administrators use advanced settings to fine-tune application behavior, set defaults, manage users, enable webinars, check quotas, and more. (475 characters)

**Edited meta description:** With this learning path, get started using Microsoft Dynamics 365 for Marketing with modules that guide you through basic setup and advanced configurations. (156 characters)

### TIP

The original meta description for this page is more than 400 characters. Write meta descriptions that fit character limits (~160 characters maximum), directly explain the page topic, use unique language from the on-page copy, and ideally include a CTA (call to action). In this case, the edited meta description describes the learning path as opposed to simply summarizing Microsoft Dynamics 365 for Marketing. The edited meta description also includes a CTA ("get started") to encourage users to click on the page and go through the modules. CTA is not required but will enhance the meta description. The introductory paragraph can often be used for the meta description by paring down the length and modifying a bit of the language.

**Example 2:** <https://docs.microsoft.com/learn/modules/intro-to-containers/>

**Current meta description:** In this module, you'll learn how to use Docker to quickly deploy a web application. You'll download an existing Docker image from Docker Hub, and customize it to run your own application. You'll upload the image to an Azure Container Registry, and run the application in Azure from an Azure Container Instance. (322 characters)

**Edited meta description:** In this module, learn how to use Docker to quickly deploy a web application, work with Docker images, and customize images to run your own applications. (153 characters)

**TIP**

In this example, the original meta description included many aspects of an ideal meta description, but it was too long. The length was reduced by removing optional phrases and tightening the overall wording. Now, the meta description will fit within character limits.

**Example 3:** <https://docs.microsoft.com/learn/modules/consume-rest-services>

**Current meta description:** Consume REST web services in Xamarin apps. (41 characters)

**Edited meta description:** In this learning module, learn how to consume REST web services with HttpClient as part of Xamarin applications. (111 characters)

**TIP**

In this case, the original meta description duplicated the H1 and was a bit too short to qualify as a full-fledged meta description. To make the meta description a bit more robust, more details were added and a clear CTA was included. The edited meta description will provide more information to users and reduce the chances that a search engine will fill in the meta description field with other on-page copy.

**Example 4:** <https://docs.microsoft.com/sccm/apps/deploy-use/create-deploy-scripts>

**Current meta description:** Create and run Powershell scripts on client devices. (51 characters)

**Edited meta description:** Learn how automated and sophisticated scripts of PowerShell can enable software deployment from configuration manager console. (143 characters)

**TIP**

The original meta description was too short and lacked detail. By using text from first paragraph and rearranging, the new meta description fits the character limit and gives an inviting message to users to click on the article.

# SEO: Tips for writing a heading 1 (H1)

11/2/2020 • 3 minutes to read

The H1 (heading 1) HTML tag is the first heading at the top of a page. The H1 has the second greatest impact on search ranking and the greatest impact on on-page relevance.

Like the title tag in metadata, the H1 must quickly communicate the intent of the article: What a customer can learn or accomplish with the article.

One of the most complicated tasks of search engines is to understand the meaning and content of a page. In order to achieve that, the crawler uses different signals, one of which is the content in H1 tag. Using relevant keywords in the H1 helps search engines understand the intent of the page.

## IMPORTANT

Don't confuse the H1 with the title tag; they are two different text elements. The H1 displays at the top of the article and has a limit of 100 characters including spaces. The title tag is in article metadata and displays in the browser bar and on the search engine result page.

## Rules for writing H1s

- Every page must have a single H1. Don't use more than one H1 per page.

It is OK to incorporate multiple H2 and H3 tags on a page.

- The H1 tag is at the top of the page and listed before any subheadings (H2 and H3 tags).
- Echo the language and intent from the title tag.

Because the H1 allows more characters than a title, it can be more descriptive.

- Ensure that every page has a unique H1 tag that is informative and specific to the on-page copy.

Don't use the same H1 on two or more pages.

- Include primary keyword(s) in the H1 tag to maximize intent and SEO value.
- Ensure that the H1 isn't generic; it needs to communicate a specific intent.
- Don't substitute bold or italicized text for heading tags.

## H1 heading examples

Example 1: <https://docs.microsoft.com/windows/win32/devices>

Conversational H1: Devices (7 characters)

Edited H1: Overview of device APIs for Windows desktop app technologies (60 characters)

## TIP

The current H1 tag is vague and overlaps with other Win32 pages. The updated H1 tag is much more informative and specific to the page, which ultimately helps search engines rank the content more effectively. One word is unlikely to ever be enough to describe the intent of a page. Add detail to sparse or very short H1 text.

**Example 2:** <https://docs.microsoft.com/windows/uwp/gaming/getting-started>

**Current H1:** Getting started (15 characters)

**Edited H1:** Get started with game development on Windows and Xbox (53 characters)

**TIP**

Like the previous example, this H1 tag is vague and generic. Any generic phrases like "get started," "how to," "reference," or "tutorial" need additional information to form a fully optimized H1 tag. In this case, added details about the topic (game development on Windows and Xbox) make the H1 tag much more informative and valuable for search engines.

**Example 3:** <https://docs.microsoft.com/azure/virtual-machines/windows/snapshot-copy-managed-disk>

**Current H1:** Create a snapshot (17 characters)

**Edited H1:** Create a snapshot of a virtual hard drive in Azure (50 characters)

**TIP**

Like the other examples, the current H1 is sparse and generic. The H1 should contain specific details about the page, so the edited H1 includes more detail for users and search engines to better understand the intent.

**Example 4:** <https://docs.microsoft.com/sql/connect/odbc/linux-mac/system-requirements>

**Current H1:** System Requirements (19 characters)

**Edited H1:** ODBC driver system requirements for SQL Server on Linux and Mac (63 characters)

**TIP**

Like one-word H1s, H1s using just two words rarely communicate page intent adequately. In this case, the current heading isn't specific enough to communicate page relevance and won't help the page rank. The edited H1 tag includes more details that are more specific to the page copy.

**Example 5:** <https://docs.microsoft.com/azure/cloud-adoption-framework/migrate/azure-migration-guide>

**Current H1:** Azure migration guide: Before you start (39 characters)

**Edited H1:** Azure migration guide: Choices to make before you start (55 characters)

**TIP**

The current H1 is detailed and relevant to the page content, so the edited version contains only minor changes. Since H1 tags don't have the same strict character limits as title tags, the language can be less clipped. Otherwise, this H1 tag is a good example of relevant, detailed, SEO-valuable content.

# What is Google Quick Answer Box

11/2/2020 • 4 minutes to read

Google quick answer boxes provide users with scannable, easy-to-digest answers at the top of the search results so that users can find the information they seek without having to click off to another website.

These answer boxes are pulled from high-ranking websites that Google trusts to provide users with the correct response. They appear most frequently in response to question queries, such as those beginning with 'what is' or 'how to'.

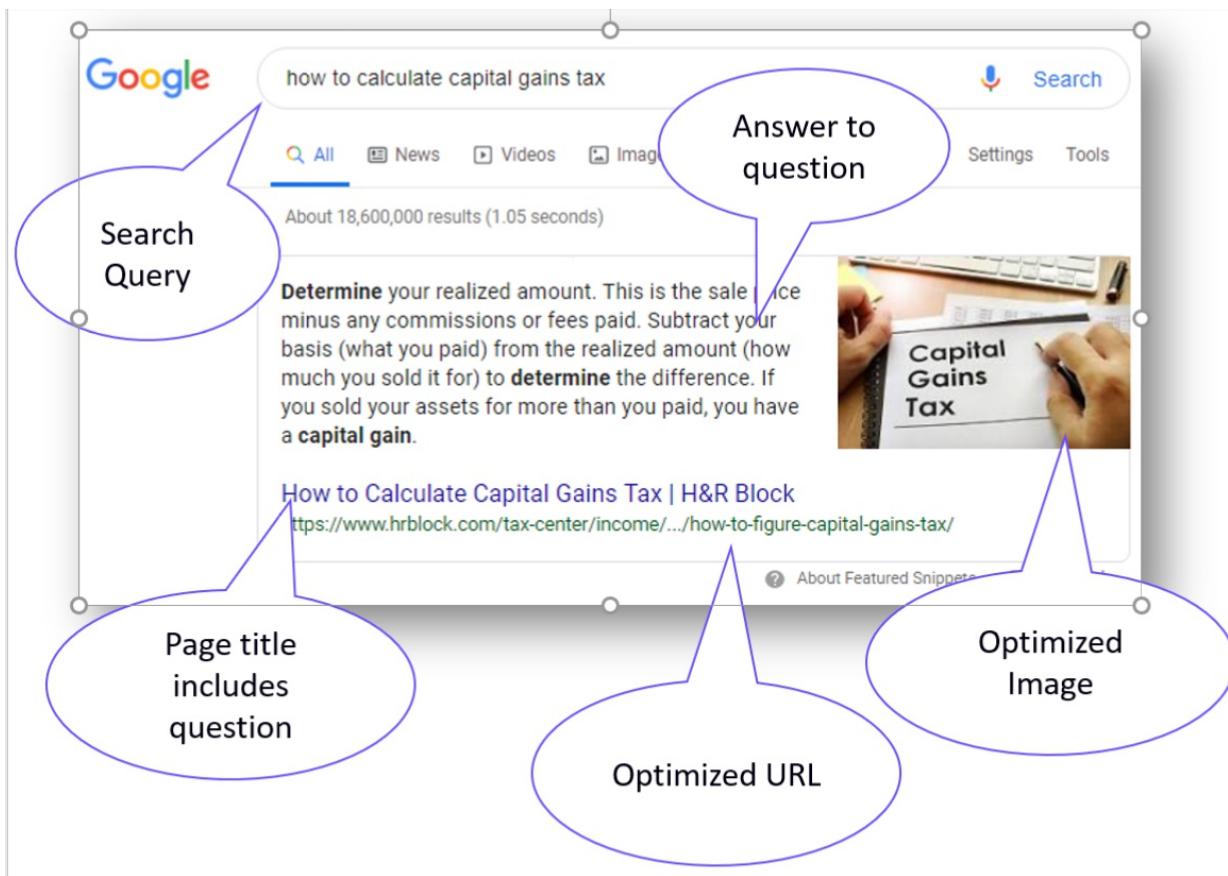
The quick answer box result allows users to get the information they need without clicking on the blue link. Long tail keywords will be an effective tool in helping product or service articles to appear in Google's rich answer box. Because rich answers focus on searched questions rather than keywords, it's important to place an emphasis on these longer phrases to yield results. Focus on long tail keywords that are question and answer oriented in order to improve your likelihood of appearing in Google's rich answer box.

## Quick Answer Box page elements best practices

Google quick answer box is also referred as position zero, which mean ranking above position 1, sometimes also called **Featured Snippet**. To make the article appear in Google Quick Answer Box, following best practices are recommended.

1. Include the question in the URL, page title and H1.
2. Answer the question directly after the H1 using around 300 characters, avoid parenthesis.
3. Include the answer in the Meta Description.
4. Include the supporting content with internal links and CTAs, keep under 2,000 words.
5. Include an image on page with optimized alt text.
6. Target keyword should rank within the top five positions in Google to receive a Quick Answer Listing.

Below image shows how different elements make a quick answer box results in Google search result.



## How to find which docs pages are ranking for Quick Answer Box in Google

Following steps will help find which Key phrases and Docs URLs are resulting in position 0 on Google Search as Quick Answer box.

1. To start the process, you will need access to BrightEdge tool. Reach out to khairunj@microsoft.com for permission.
2. Accept the invite from [BrightEdge](#).
3. Once logged into BrightEdge, the platform will take you to the BrightEdge dashboards home page.
4. Click on the gray side bar to expose the different features of the tool.
5. Click on **Data Cube**, sometimes you might not find the DataCube on the side bar and click on DataCube in Sidebar Navigation, it's possible that DataCube might not be listed on side bar. If that happens click on **Research** and then on **Content Research** to be taken to DataCube.
6. Enter the URL to Research, the URL can be the entire domain, a specific page URL or a subdirectory. Click on the Search button.
7. To better understand the specific types of content shown in the Search Engine Result Page as quick answer box navigate to **Content Strategies** tab.

**BRIGHTEDGE** Data Cube

Leverage the industry's largest and actionable content repository to develop winning strategies for your business.

Jan 2020 - Jun 2020 (Monthly)

Overview High Ranking Keywords Long Tail Keywords Content Strategies Site Comparison Filters

Total Universal Results: 59,238 (↓21%) 25,814 (↓36%) 81 (↓16%) 606 (↓5%) 0 Places 5,439 (↓7%) 0 Local 3-Pack 27,298 (↓3%) People Also Ask

Content Strategies

Universal Results

**Total Universal Results**

Category	Value
Images	33.1K
Videos	16.8K
Carousel	81
Places	606
Local 3-Pack	0
Quick Answers	5,439
People Also Ask	27,298

## Content Strategies - high-level results

Content strategies will display all types of results for docs base URL. To get results only related to Quick Answer Boxes, click on the Quick Answers link. The above image shows that 5,439 **Quick Answers** for Docs site at that time of this article creation.

## Content Strategies - Quick Answer results for specific site section

If you are a content developer in SQL team and want to only see the Quick Answers for SQL content just enter <https://docs.microsoft.com/sql/> and click on Search button. Click on Quick Answers Link and in results pane you will find all the Quick Answers for SQL URLs.

**BRIGHTEDGE** Data Cube

Leverage the industry's largest and actionable content repository to develop winning strategies for your business.

Jan 2020 - Jun 2020 (Monthly)

Overview High Ranking Keywords Long Tail Keywords Content Strategies Site Comparison Filters

Total Universal Results: 4,546 (↓15%) 1,154 (↓39%) 0 Videos 6 (↑77%) 0 Places 632 (↑10%) 0 Local 3-Pack 2,754 (↓4%) People Also Ask

Content Strategies

Universal Results

**Quick Answers**

Category	Value
Images	1,154
Videos	0
Carousel	6
Places	0
Local 3-Pack	0
Quick Answers	632
People Also Ask	2,754

https://app18.brightedge.com/ui/rp/11/data\_cube\_research

**BRIGHTEDGE** Data Cube

Track Columns

Keyword	Search Volume	Blended Rank	Page	Blended Rank Change	Category
: subquery in sql	8,100	1	docs.microsoft.com/en-us/sql/relational-databases/perform...	100 ↑	Quick Answers
: subquery sql	8,100	1	docs.microsoft.com/en-us/sql/relational-databases/perform...	100 ↑	Quick Answers
: sql contains	6,600	1	docs.microsoft.com/en-us/sql/t-sql/queries/contains-transact...	100 ↑	Quick Answers
: aggregate function sql	5,400	1	docs.microsoft.com/en-us/sql/functions/aggregate-funct...	100 ↑	Quick Answers
: aggregate functions in sql	5,400	1	docs.microsoft.com/en-us/sql/t-sql/functions/aggregate-funct...	100 ↑	Quick Answers
: aggregate functions sql	5,400	1	docs.microsoft.com/en-us/sql/functions/aggregate-funct...	100 ↑	Quick Answers
: indexing in sql	5,400	1	docs.microsoft.com/en-us/sql/relational-databases/indexes/cl...	100 ↑	Quick Answers
: indexing sql	5,400	1	docs.microsoft.com/en-us/sql/relational-databases/indexes/cl...	100 ↑	Quick Answers

Click on Export icon and then the CSV export option to get a raw file of all of the keywords and URLs ranking for Quick Answer Boxes.

To know which keywords are taking position zero (Quick Answers) from your content competitors URLs do the above steps by entering competitors URL and browse through their Quick Answers. From there, you can build out a strategy to take over their quick answer box results.

### Content Strategies - Quick Answer Box Opportunities

1. To find new quick answer box opportunities, use the High Ranking Keywords report.
2. Click on Ranked on Pos. 2-5 to find keywords that could potentially be optimized to garner a quick answer box result.
3. Use filters to set category as "Regular Web Listing."
4. Review keywords and make use of filters to determine which queries may be an "easy win" in achieving position "0."

#### NOTE

Not all queries trigger a quick answer box listing. Always conduct a Google search prior to updating your content to go after answer boxes.

The screenshot shows the BrightEdge Content Performance interface. The left sidebar includes links for Anomalies, Home, StoryBuilder, Keyword Reporting, Page Reporting, Data Cube (which is selected), Recommendations, Site Report, Campaigns, and More. A banner at the bottom left says 'Redesigned Navigation & Dashboards' and 'See What's New →'. The main content area has a title 'High Ranking Keywords' with a dropdown set to 'United States - E' and a URL 'https://docs.microsoft.com/en-us/sql/'. Below this are four summary boxes: 'Overview' (15,001 +2%, 2,693 +16%, Total Organic Keywords), 'High Ranking Keywords' (8,754 +3%, Ranked on Pos. 1-5), 'Long Tail Keywords' (3,554 +5%, Ranked on Pos. > 5), and 'Content Strategies'. A chart titled '# Keywords Ranked on Pos. 2-5' shows a green bar reaching nearly 10K. Below the chart is a table titled 'Month of Jun 2020 (compared to Month of May 2020)' with 'Select Different Periods' options. The table columns are Track, Keyword, Search Volume, Blended Rank, Page, Blended Rank Change, and Category. Several rows are highlighted with yellow boxes, including 'sql service', 'insert sql', 'sql insert', 'sql inserted', 'sql server management studio', 'sql server management studio', and 'sql server manager studio'. The 'Category' column for these rows shows 'Regular Web Listing', 'Regular Web Listing', 'Regular Web Listing', 'Regular Web Listing', 'Site Links', 'Site Links', and 'Regular Web Listings' respectively.

In above example "*sql insert*" and "*insert sql*" can be good candidate to move to position 0 from position 5 by making some changes to the page elements.

## Next Steps

- Track keywords in STAT analytics tool [Docs search keywords research and tracking](#) and measure the ranking performance.
- Learn more about [What is a Keyword or Key Phrase?](#)
- Refresh yourself on [SEO Basics](#).

# What is a Keyword or Key Phrase?

11/2/2020 • 5 minutes to read

A keyword is any query people use in search engines. A keyword can be one or more words. It's how users attempt to answer their questions or seek a way to resolve an issue. It is how the user expresses their intent.

## How to find keywords your article is ranking on in Google

There are a number of tools that will help you discover which keywords are causing your article to be displayed in search engine results. Understanding what your article is currently ranking on is the first step. It doesn't answer the question of what keywords your article could or should be ranking for across all possible audiences.

The tools in this article provide Google US keyword data. This data is useful, because most of our content is written in US English. Google drives 90% of search traffic to the Microsoft Docs website.

These tools will also allow you to get data for individual pages or for all articles on a URL path.

Google search console, BrightEdge and SEMRush are used to perform the Keywords research for new and existing articles. STAT Analytics is designed and used for tracking the keywords for ranking overtime.

To gain access to Google Search Console, STAT Analytics and BrightEdge and SEMRush, please contact [Khairun Jamal](#). Provide Google account credentials for access to Google Search Console and your Microsoft alias for access to BrightEdge.

## When to use which keyword research tool

Following comparison should help determine which tool to use for a specific keywords research purpose.

GOOGLE SEARCH CONSOLE	BRIGHTEDGE	SEMRUSH	STAT ANALYTICS
Provides the list of keywords the article ranked in Google	Provides search volume data for keywords	Provides search volume data for keywords	STAT is used to track the keywords ranking and search volume data
Google impressions and clicks for site section and articles	Provides data for <b>Long tail</b> and <b>high ranking</b> keywords	Provides <b>keywords variations</b> and its search volume	STAT provides <b>related searches</b> reports from Google
Average position on Google for the site	Provides data for <b>backlinks</b> and <b>Quick answer box</b>	Provides <b>related keywords</b> and its search volume data	STAT provides "people also ask" reports from Google.

### NOTE

SEMRush is an advanced keyword research tool and C&L has only limited licenses.

## Using Google Search Console

Google Search Console is useful for identifying the queries that users are typing into the Google search engine to find your existing article or a similar topic for your product or service.

The screenshot shows the Google Search Console interface for the URL <https://docs.microsoft.com/en-us/azure/cognitive-services/>. The left sidebar is collapsed, and the main content area displays the 'Performance on Search results' report.

- Section 1:** URL inspection bar at the top, showing the URL and a search input field.
- Section 2:** Left sidebar with 'Search results' selected (highlighted with a red box).
- Section 3:** Performance metrics summary and line chart. Metrics shown: Total clicks (205K), Total impressions (3.21M), Average CTR (6.4%), and Average position (15.6). Below this is a line chart showing Clicks and Impressions over time from June 15, 2019, to September 7, 2019.
- Section 4:** Detailed query performance table. The table has columns: QUERIES, PAGES, COUNTRIES, DEVICES, and SEARCH APPEARANCE. It lists two queries:
 

Query	Clicks	Impressions
azure cognitive services	796	18,304
azure ocr	526	3,524

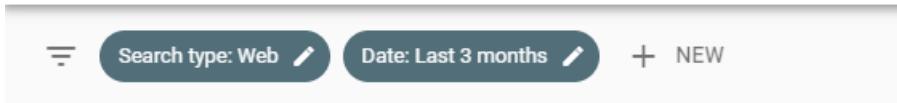
1. The property that is being measured.
2. You want to have Search Results selected.
3. Where you set the metrics and filters for measurement.
4. Where your query results are detailed.

In the example provided, there is a pre-defined property in the console: cognitive-services. When requesting access to the console, you can request a property to be defined for your use or you can use the filters to refine your results.

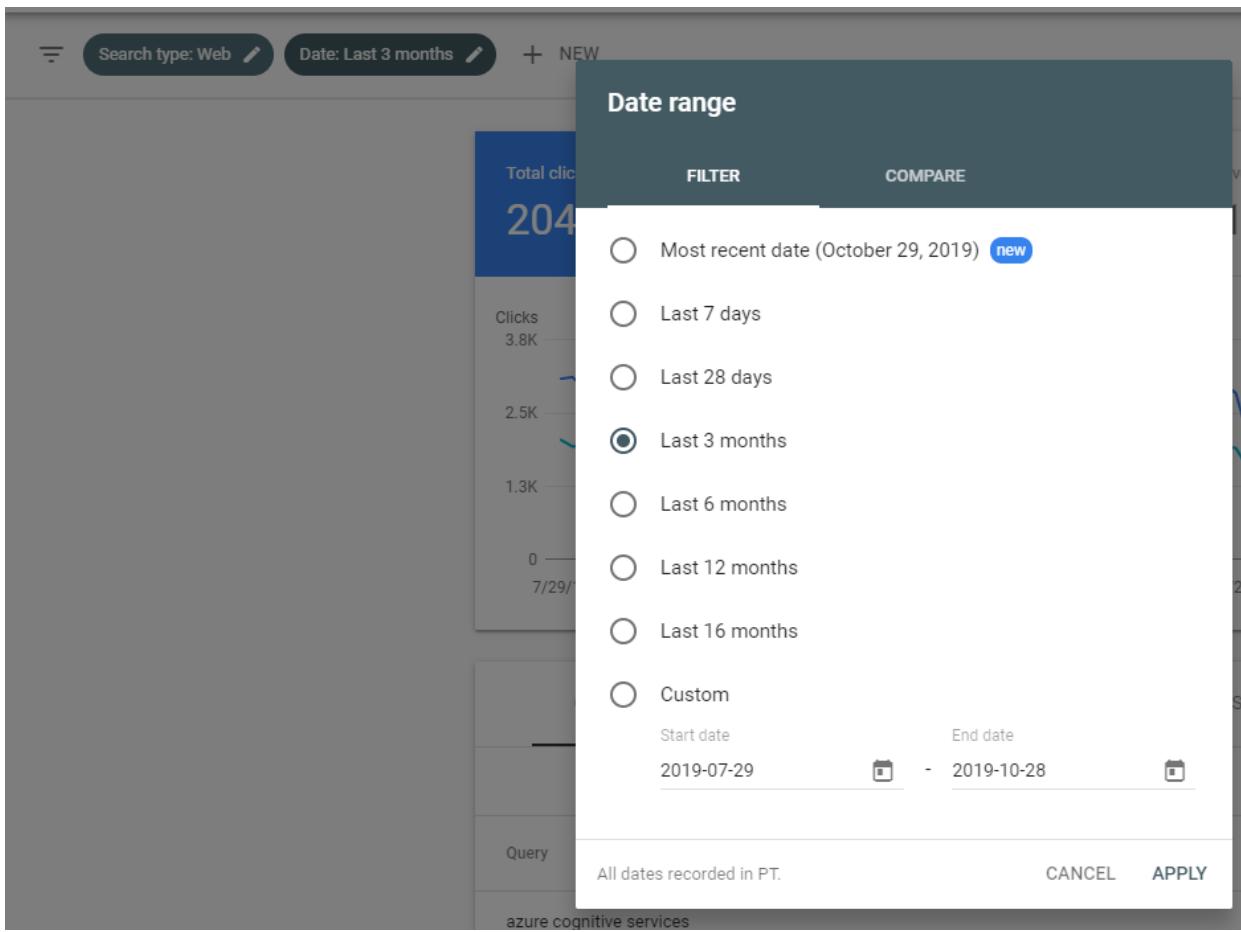
## Configuring the Google Search Console report

The default metrics that will be set upon entry will include:

- Search type: Web
- Date: Last three months

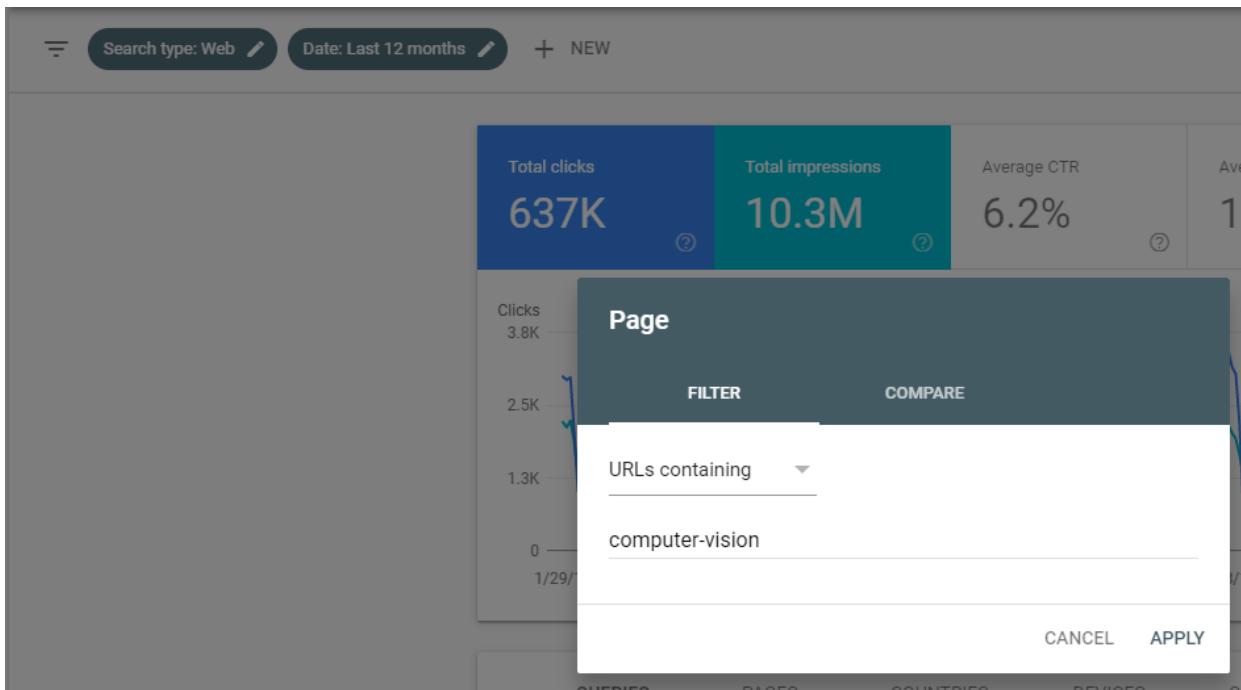


Each metric can be edited by selecting the pencil icon on the metric label.



You can select new metrics including query, page, country, device, and search appearance. For the purpose of this article, we will focus on the page and query metrics.

Select the "NEW" button and choose "Page." Type or paste the URL of a page or a unique phrase like your service or product name as it appears in the URLs in your doc set. Select "Apply."



Your results will be displayed. Your top results will most likely include branded terms. You'll want to dig deeper into the results to find non-branded terms with significant traffic but not necessarily high click count.

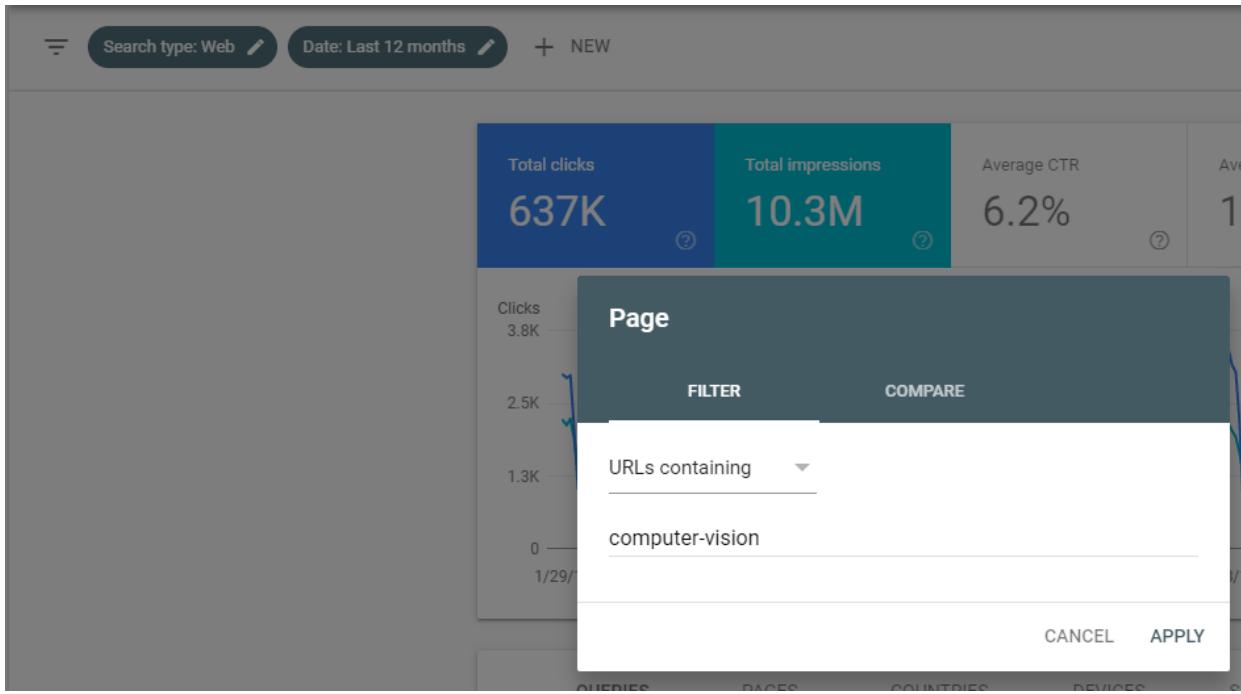
The screenshot shows a table of search queries with columns for Query, Clicks, and Impressions. The rows include various terms related to computer vision and Microsoft services.

Query	↓ Clicks	Impressions
azure ocr	1,344	43,583
computer vision api	719	25,397
azure computer vision	553	33,168
microsoft computer vision	444	26,891
microsoft vision api	439	21,601
azure object detection	388	2,243
microsoft computer vision api tutorial	328	8,185
microsoft computer vision api	322	13,178
azure ocr api	316	13,330
microsoft ocr api	305	13,746

Rows per page: 10 1-10 of 1000 < >

Mining these results will help you identify key words and phrases you aren't optimizing for but your potential users are searching for on the web.

You can also use this tool to see how well your content is doing against specific key words and phrases. Select the "NEW" button and choose "Query." Type or paste the key word or phrase you want to explore. Select "Apply."



Sort the results by the number of impressions. The phrases with the highest number of impressions but low clicks may be a good starting point for you to explore new key word and phrase ideas.

## Using BrightEdge

The BrightEdge Data Cube provides a 12-month average search volume based upon data that the service acquires from Google. By providing an average, the service helps you understand how your content performs over time, without the effects of seasonality.

To get started, select "Research" in the left-hand navigation and "Content Research" in the flyout menu. Type or paste the URL of the doc set that you want to research keywords for in the box at the top of the screen.

The screenshot shows the BrightEdge Content Research interface. The left sidebar includes links for Home, Dashboards, Research (selected), Action, Reports, Help, Settings, and a user profile for Terry Christiani. The main content area has a header "Content Research POWERED BY DATA CUBE" and a sub-header "Perform keyword and competitive strategy research based on a Domain, URL or Keyword". It features a search bar and a date range selector "Oct 2018 - Sep 2019 (Monthly)". Below this are tabs for Overview, High Ranking Keywords, Long Tail Keywords, Content Strategies, and Site Comparison. Key performance metrics are displayed: Data Cube Score (322, +5%), Total Organic Keywords (226, +13%), Ranked on Page 1 (83, +9%), Ranked on Page 2 (27, +80%), and Ranked on Page 3 (20, +43%). A chart titled "Content Performance" shows the Data Cube Score over time from Oct 18 to Sep 19. A table titled "Top 5 Keywords" lists the most searched keywords with their ranks, blended rank change, search volume, and category. The table data is as follows:

Keyword	Rank	Blended Rank Change	Search Volume	Category
azure colored	38	63 ▲	12,100	Regular Web Listing
azure colour	26	75 ▲	9,900	Regular Web Listing
image tagging	21	5 ▼	5,400	Regular Web Listing
image tags	57	20 ▲	5,400	Regular Web Listing
tags image	52	14 ▲	5,400	Regular Web Listing

By default, BrightEdge shows keyword results for everything on a given URL path. For example, for this URL: <https://docs.microsoft.com/azure/cognitive-services/>, BrightEdge will provide data for all pages that have that URL at their root. You can filter on an individual page using the Filters feature. For a URL that ends in a file name, only the specific page data is given.

### Reading through the BrightEdge results

The report will default to an overview of your content's performance for the last 12 months. It will also provide you with the top five keywords that your URL is ranking for in this period.

The high ranking keywords tab will display, in the order of their respective search volume, keywords where your URL rank on the first page of results.

The long tail keywords tab will display key phrases longer than three words where your URL is being served as an impression to the user. This will provide you with results for page 1 through page 10 in search results.

The content strategies tab shows how your URL ranks among different media and areas within Google search results including images, videos, carousel, places, local 3-pack, quick answers, and people also ask.

The site comparison tab allows you to compare your URL to a competitor's URL. It will show you how your content performs against the competition and provide you with a keyword gap analysis. The gap analysis allows you to see what words or phrases are working for your competition. Non-branded terms in this list can and should be considered for use in your articles.

Compare against: https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html

Content Performance Keyword Gap Analysis

Displaying keywords ranked on https://docs.aws.amazon.com/rekognition/latest/dg/what-is.html but not on https://docs.microsoft.com/computer-vision/

Month of Sep 2019 (compared to Month of Aug 2019) Select Different Periods

Keyword	Blended Rank	Blended Rank Change	Page	Search Volume	Category	Mobile Friendly
amazon rekognition	2	No Change	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	2,900	Site Links	N/A
aws rekognition	2	2 ▲	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	1,900	Site Links	N/A
aws image recognition	4	97 ▲	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	210	Site Links	N/A
amazon rekognition sample	5	96 ▲	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	70	Regular Web Listing	N/A
aws rekognition face recognition	25	8 ▼	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	40	Regular Web Listing	N/A
aws rekognition documentation	4	No Change	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	30	Site Links	N/A
amazon rekognition example	6	2 ▼	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	20	Regular Web Listing	N/A
amazon rekognition logo	4	No Change	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	20	Regular Web Listing	N/A
how does amazon rekognition work	5	No Change	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	20	Regular Web Listing	N/A
amazon rekognition for images	7	11 ▲	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	0	Regular Web Listing	N/A
aws rekognition fees	8	93 ▲	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	0	Regular Web Listing	N/A
visual recognition wiki	64	2 ▼	docs.aws.amazon.com/rekognition/latest/dg/what-is.html	0	Regular Web Listing	N/A

Show 25 entries Showing 1 to 12 of 12 entries First Previous | Next Last

## Next Steps

- Track keywords in STAT analytics tool [Docs search keywords research and tracking](#) and measure the ranking performance.
- Refresh yourself on [SEO Basics](#).
- Gather your colleagues together for a [SEO Hackadoc](#).

# Docs search keywords research and tracking

11/2/2020 • 9 minutes to read

Keyword research is one of the best ways to understand and reach customers from organic search. Keywords truly represent the voice of the customers, so it's important to identify the most valuable terms that users are searching for. This can include broad, competitive keywords, long-tail, informational searches, branded queries, and non-brand terms. By identifying a large swath of relevant keywords with search volume, a writer can more meaningfully create content, optimize existing content, and track content for performance. Keyword research also helps to inform overall content strategy, including content development, gap opportunities and content prioritization.

## Why is it important to track keywords in STAT analytics

Once the keyword research process is complete, it's important to track these terms in STAT analytics to understand overall content performance which means monitoring the ranking in Google. Which docs pages are performing well, and where do we have gap and opportunities exist? Is the docs content performing better for brand or non-branded queries? What about long-tail vs. broad, competitive keywords? Keyword rankings provide the data that can lead to strategic actions. These strategic actions can boost organic rankings, site visibility and visits.

## Keyword research process

There are 5 steps for the keyword research process. It's recommended to work through this process by different site section:

1. Request access to the STAT Analytics and BrightEdge Data cube
2. Examine existing keywords in STAT
3. Select keywords from the "People Also Ask" and "Related Searches" Reports in STAT
4. Select keywords from the BrightEdge DataCube tool
5. Upload the filtered keywords list in STAT.

Since the keyword tracking limit in STAT analytics tool is about ~10,000 keywords for Microsoft Docs, it's recommended to track about 200-400 keywords for main site section (Azure, Windows, SQ, Visual Studio). Depending on business priority, the size of each site section, or upcoming content initiatives, some site sections may need more or less tracked keywords. It's also recommended to track all keywords in the Keyword Universe, which is listed in the folder "MS Docs – KW Universe".

### Step 1: Get access to STAT analytics and BrightEdge

To gain access to STAT analytics send your email to khairunj@microsoft.com and she will grant the access to all the tools to start the keyword research.

### Step 2: Examine existing keyword in STAT

After access is granted, go to STAT analytics tool at <https://msazure.getstat.com/login> and find the tag(s) for your site section under "MS Docs – KW Universe". Click on the "Microsoft Docs – all". Expand the arrow next to "Site" to see all of the tags listed under DataViews . Depending on your access you might be able to see the other projects created by different teams.

## Projects

- ▷ AI and Machine Learning
  - Azure Government
  - Azure Media Services
  - ▷ Azure Notebooks
    - Azure Security
      - Azure Video Indexer
  - ▷ Big Data
    - Identity
    - Infer.NET
  - ▷ Java dev center
  - ▷ ML.NET
  - ▷ ML.NET-API
- ◀ MS Docs - KW universe
  - ▷ Azure
  - ▷ Learn
    - Microsoft Docs - all
- ▷ MS Docs - Power BI
- Node dev center
- ▷ Patterns & Practices
- ▷ Python Dev Center
- ▷ SQL Server
- ▷ Visual Studio & VSTS

Once you find the tag(s) for your site section, click on the tag(s) and download the table(s) to get a list of all the keywords. Right click the Keywords table and click on "Export Table" to download a CSV file.

The screenshot shows the STAT Data Views interface with the 'Keywords' tab selected. On the left, there's a sidebar with a tree view of site sections, including 'aspnet' which is currently selected. The main area displays a table titled 'Keywords Dec 19 (Data for the tag aspnet)'. The table has columns for 'Keyword', 'Rank', 'URL', and 'Avg'. A context menu is open over the table, with the 'Export Table' option highlighted by a red box. Other options in the menu include 'Tag Keywords', 'Untag Keywords', 'Delete', 'Toggle Tracking', and 'Keyword Duplicator'.

Keyword	Rank	URL	Avg
api asp net	3	docs.microsoft.com/en-us/aspnet/web-api/overview/tutorials/getting-started-with-aspnet-web-api/tutorial-your-first...	0
asp core	1	docs.microsoft.com/en-us/aspnet/core/	2400
asp dot net	1300		
asp net	3	docs.microsoft.com/en-us/aspnet/	60500
asp net 5	3	docs.microsoft.com/en-us/archive/msdn-magazine/2014/special-issue/asp-net-5-introducing-the-asp-net-...	580
asp net code	720		
asp net core	1	docs.microsoft.com/en-us/aspnet/core/	40500
asp net core identity	1	docs.microsoft.com/en-us/aspnet/core/security/authentication/identity	4400
asp net core mvc	1	docs.microsoft.com/en-us/aspnet/core/mvc/overview	6600
asp net core mvc tutorial	2	docs.microsoft.com/en-us/aspnet/core/tutorials/first-mvc-app/start-mvc	1300
asp net core tutorial	3	docs.microsoft.com/en-us/aspnet/core/	6600
asp net framework	2400		
asp net identity	1	docs.microsoft.com/en-us/aspnet/core/security/authentication/identity	3600
asp net mvc	3	docs.microsoft.com/en-us/aspnet/mvc/overview/introduction/getting-started	33100

Next, look through the keywords listed in the CSV file and ask the following questions to determine whether the keywords should continue to be tracked or removed.

- Is the keyword currently ranking well for a page?
- Is the keyword relevant to the site section? If users search for the term(s), should they expect to see Docs listed on the first page of the search engine results?
- Does the keyword have search volume?

### IMPORTANT

All keywords do not necessarily need to rank in Position #1 or the first page to be valuable for tracking. If the keyword has the potential to rank well, or has value for the site section, then the keyword should stay in the STAT platform, even if it isn't currently performing well. Also review which keywords have very high volume and docs site is not ranking to find the opportunities.

## Step 3: Select keywords from the People Also Ask and Related Searches Reports in STAT

Go to the "MS Docs – KW Universe" project in STAT and click on "Reporting" under Site Tools.

## IMPORTANT

You need to be in the correct project to pull reports that will be most valuable. You can also pull these reports from other projects that are relevant to your site section.

- ▷ Site (1435)
- ◀ Azure Docs
  - azure docs - active directory (30)
  - azure docs - analysis services (1)
  - azure docs - app service (4)
  - azure docs - application gateway (1)
  - azure docs - architecture (18)
  - azure docs - aspnet (1)
  - azure docs - automation (10)
  - azure docs - availability zones (1)
  - azure docs - avere (1)
  - azure docs - azure cli (1)
  - azure docs - backup (7)
  - azure docs - batch (3)
  - azure docs - billing (5)
  - azure docs - bot service (2)
  - azure docs - cdn (2)
  - azure docs - cloud services (5)
  - azure docs - cognitive services (3)
  - azure docs - competitors (8)
  - azure docs - connectors (1)
  - azure docs - container service (1)
  - azure docs - cosmosdb (9)
  - azure docs - cost management (1)
  - azure docs - data catalog (1)
  - azure docs - data factory (2)
  - azure docs - data lake (4)
  - azure docs - data profile (1)
  - azure docs - data services (5) [highlight]
  - azure docs - data sharing (1)
  - azure docs - databricks (1)
  - azure docs - developer (1)
  - azure docs - devops (31)
  - azure docs - disaster recovery (3)
  - azure docs - dns (2)
  - azure docs - dotnet (1)
  - azure docs - eventhub (1)
  - azure docs - expressroute (1)
  - azure docs - firewall (2)

## Site Tools

- Add Keywords
- Alerts
- Reporting [highlight]
- Settings

Once you click on the Reporting tab, go to "Create Report" to pull the People Also Ask and Related Searches Reports.

The screenshot shows a reporting interface with the following elements:

- Top navigation bar: Download Reports, Manage Scheduled Reports.
- Main toolbar: Create Report, Download Selected, Delete Selected.
- Table view:

Title & Description	Report Type	Contents
Azure Docs - People Also Ask	Keyword: People also a	Site: Microsoft Docs - all Tag: azure
Related Searches - Azure Docs	Keyword: Related sear	Site: Microsoft Docs - all Tag: azure

There will be a series of steps to pull the **People Also Ask** and **Related Searches Reports**. For People Also Ask, select "**People also ask (Google)**", and for Related Searches, select "**Related searches (Google)**". Once you click the report that you want to download, click "**Next**" on the bottom right-hand side. From there, you will

name the report and set an address to receive a confirmation e-mail once the report is ready to download.

The next step will allow you to specify which keywords to pull the report for. Click "All keywords tagged with" and include the tag that matches your site section. If you want to pull a report for multiple tags, you will



pull a separate report for each tag.

Next, specify the date range for your report and how often to run the report. For this process, you can click "**Once only**" and specify a date range of this month. For the final step, you can include global and regional search volumes. Click "**Finish**" on the bottom right-hand side to run the report. You will follow the same process for the People Also Ask and Related Searches reports.

Once you pull the reports and download the CSV files, remove duplicates and make the files a bit easier to digest. The report will show results for every day in the date range, so follow the steps below to remove duplicates results:

**People Also Ask Report:** In the Excel spreadsheet, go to "Data" - "Remove Duplicates" and specify Column "PAA Question" only. This will remove all of the duplicate People Also Questions listed in the spreadsheet.

**Related Searches Report:** In the Excel spreadsheet, go to "Data" - "Remove Duplicates" and specify Column "Related Searches" only. This will remove all of the duplicate Related Searches in the spreadsheet. From there, look at the columns "PAA Question" for the People Also Ask report and "Related Searches" for the Related Searches report to assess the end users keywords choices for searching. Use the same general thought process as Step #2 to determine if the keyword(s) should be added to STAT for tracking, Is the keyword relevant to the site section? If users search for the term(s), should they expect to see Docs listed on the first page of the search engine results. Sift through the reports and identify any keyword(s) of interest for tracking. These results will likely be more long-tail, possibly more non-brand and a bit more niche to expand your keyword portfolio. If the keyword lists are too long/cumbersome to sift through term by term, try filtering the specified columns with the following to home in on important terms:

- Keywords containing "documentation"
- Keywords containing "vs" or "comparison"
- Keywords containing "tutorial"
- Keywords containing "what is"
- Keywords containing "definition"
- Keywords containing "how to" or "how do"

This is just a starting point, but it can help to narrow down the list a bit.

#### **Step 4 - Select keywords from the BrightEdge DataCube tool**

## NOTE

This step is very important if currently there are no existing keywords tracked in STAT.

In BrightEdge, go to the DataCube tab on the left-hand side. You will see a field with a place to enter a URL. Make sure that the search engine setting is "United States - English" on the left-hand side of the search bar and click on Search.

The screenshot shows the BrightEdge interface with the 'Data Cube' tab selected. In the search bar, the URL 'https://docs.microsoft.com/en-us/aspnet' is entered. Below the search bar, there are four categories: 'Total Organic Keywords' (1,864), 'Ranked on Page 1' (355), 'Ranked on Pos. 2 - 5' (906), and 'Ranked on Pos. > 5' (603). The 'Ranked on Page 1' section is highlighted with a yellow box.

Once you enter the URL for example <https://docs.microsoft.com/aspnet> you will see a number of results for Total Organic Keywords, Ranked on Page 1, Ranked on Page 2, and Ranked on Page 3. Go to "Ranked on Page 1" to hone in on all keywords currently ranking on Page 1. Additionally, make sure that the date range is set to the latest month possible. Next, scroll down to the listed keywords and click the arrow icon on the top right-hand side to download the keywords.

The screenshot shows the 'Track' view of the Data Cube interface. It displays a table of keywords with the following columns: Keyword, Blended Rank, Blended Rank Change, Page, Search Volume, and Download (with CSV and Excel options). The table includes rows for various search terms like 'on to facebook', 'n in for facebook', etc., along with their respective search volumes and page rankings.

Keyword	Blended Rank	Blended Rank Change	Page	Search Volume	Download
on to facebook	11	90 ▲	docs.microsoft.com/en-us/aspnet/core/security/authentication/	30,400,000	Images
n in for facebook	9	92 ▲	docs.microsoft.com/en-us/aspnet/core/security/authentication/	201,000	Images
ebook login in facebook	11	6 ▲	docs.microsoft.com/en-us/aspnet/core/security/authentication/	165,000	Regular Web Listing
ogin facebook	9	92 ▲	docs.microsoft.com/en-us/aspnet/core/security/authentication/	165,000	Regular Web Listing
ebook log in sign up	11	58 ▲	docs.microsoft.com/en-us/aspnet/core/security/authentication/	33,100	Regular Web Listing
jquery	12	No Change	docs.microsoft.com/en-us/aspnet/ajax/cdn/jquery-ui/cdnjquery	22,200	Regular Web Listing
t html	1	100 ▲	docs.microsoft.com/en-us/aspnet/web-pages/videos/introducti	22,200	Carousel

The spreadsheet will include many thousands of keywords, so it's important to clean up the spreadsheet a bit to make the data easier to sift through. First, remove all duplicate keywords (in column A) by going to "Data" – "Remove Duplicates" and only click "Column A". This will remove all duplicate keywords in the spreadsheet. Next, filter for search volumes below 1,000 and remove those keywords. Due to the sheer amount of keywords listed, it's best to focus on the highest-search volume keywords that are ranking on Page 1 first.

As general guidance, the recommendation is to stick with Page 1 Rankings when downloading information from DataCube. In terms of general guidance for search volume, I would start with 1,000+ searches for more common product and technologies. If that cutoff takes out too many keywords, then try 500+, then 250+, etc. until you find the right amount of keywords to sift through. If a site section is pretty small and not ranking for a ton of Page 1 Rankings, then you can expand and export All Keyword Rankings to include lower ranked terms. You are likely looking for ~100-150 keywords to track from the DataCube exercise.

With the DataCube report, you will be able to see search volume, current rankings, and ranking pages, so this data should really help to inform your decisions

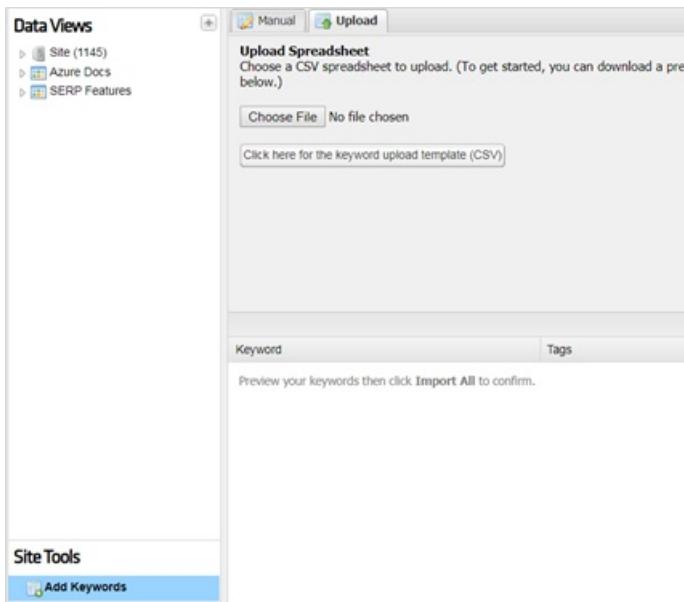
The goal number of tracked keywords is 200-400 total per site section, so it's important to whittle down the lists to the best keyword selections possible.

#### NOTE

You can also go to Google Search Console to find the keywords for your site section. [Get keywords and search rank.](#)

### Final Steps – STAT Keyword Organization and Uploading

Once the keyword list has been finalized after combining it from all sources and removing the dupes for tracking, it is important to properly tag the keywords for your site section. Every keyword should have a tag. This will make it much easier to find keywords and pull reports in the platform. You can pull reports by tag or by a group of tags (called a Dataview) that you can create in the platform. It's recommended to use the following naming convention for tags: (Site Section) – (Product Name/Group) For example, a good tag name would be Azure Docs – SQL or Azure Docs – Virtual Machines. Make sure that the tags follow the same structure so keywords are easy to find by site section and product grouping/category. Once the tagging is complete, upload all keywords to STAT in the Keyword Universe. Go to the Keyword Universe Project and click "Add Keywords" under Site Tools.



Next, go to the "Upload" tab and download the Keyword Upload Template. Fill out the CSV file with the keywords, tag names, market (US-en) and device (desktop), then upload the CSV file into the platform. You will see the keywords populate in the platform within a couple of minutes.

#### IMPORTANT

You must save the file as a CSV in order to upload the keywords into STAT. The SEO team can help with any questions or issues with the upload process.

If you are interested in creating a Dataview for your tags (by site section), please reach out to the SEO team [SEO team](#) for assistance.

# File name and path guidelines

4/9/2021 • 7 minutes to read

A strict file naming convention helps to clearly identify articles and contributes towards discoverability on the web. For most content, file and folder names are included in the public URL on docs.microsoft.com, so choose them carefully. This article provides general naming guidelines.

## IMPORTANT

Use this file name guidance for **new** articles. You don't need to rename and redirect existing articles.

## Name your files

Follow these rules for file naming:

- File names can contain **only** lowercase letters, numbers, and hyphens. Don't use spaces or punctuation characters.
- Words and numbers in the file name **must** be separated using hyphens.
- Use no more than 80 characters, which is a publishing system limit.
- Use action verbs that are specific, such as develop, buy, build, and troubleshoot. Don't use "-ing" words.
- Don't use unnecessary small words, such as a, or, and, the, and in.
- Most files must be in markdown and use the .md file extension. Exceptions are when you use .yml files for landing pages, such as *toc.yml* and *index.yml*.
- Avoid redundancy with other segments of the file path.
- Avoid unapproved or unnecessary acronyms. For Azure specifically, don't use rm or arm as acronyms anywhere in a file name.

For Image file names use the guidelines at [image file and SEO](#).

## Pattern

The general pattern for a file name is: `platform-language-content-product-version.md`

Use the parts of the pattern that apply. Review the list of articles in the repository to get an idea of existing names.

## Standard examples

Here are a few examples of valid file names that follow this pattern:

- *dotnet-continuous-delivery.md*
- *ios-get-started.md*
- *manage-account.md*
- *install-windows-server-2008r2.md*
- *azure-sdk-dotnet-release-notes-2-8.md*

The service names aren't in the file name examples, but are an element in each URL path:

- /azure/cloud-service/dotnet-continuous-delivery
- /azure/mobile-services/ios-get-started
- /azure/cosmos-db/manage-account

# Advanced file naming tasks

Here are some other things you can consider when naming files:

## Change case in file names

Windows operating systems are case insensitive. If you need to change a file name to fix casing, it's better to make a substantive change, unless you can make the change on a Linux or Mac. For example:

administration-and-Development-Task-List-in-BizTalk-Services --> services-administration-and-development-task-list

Use the following command to rename a file:

```
git mv <articles/service-folder/current-file-name.md> <articles/service-folder/new-file-name>
```

## A/B testing

A/B testing on docs.microsoft.com requires one file in the A/B pair to include `.experimental`. in the file name. For example, you would compare `file-name.md` to `file-name.experimental.md`.

## Azure Marketplace content

To distinguish content that focuses on partner contributions to the Azure marketplace, start the file names with "marketplace". This content isn't too common, as most partner content should be created on the partners' own web sites. For example:

- `marketplace-mongodb-virtual-machines-install-windows-server-2008r2.md`

## File name approval

It's the job of the pull request reviewers to review file names when a new file is submitted to the repository. If a file name must be changed, reviewers will provide feedback using the pull request comment stream. The file name criteria must be met before the pull request can be accepted for merge.

# Name your subfolders

Each subfolder in your documentation repo will generally appear as a path segment in the published URL. They should work with the [base URL](#) and individual file names to build a meaningful hierarchy. For example, subfolders might divide a large product or technology into subtechnologies or services. You can also use subfolders to group articles by type, such as `security` or `deployment`.

### NOTE

Service names are in the URL path element that follows the brand, and aren't needed in the file name.

Here are some examples:

- A `mobile-services` folder within the `azure` docset:  
`https://docs.microsoft.com/azure/mobile-services/dotnet-backend-get-started-settings-sync`
- A `mail-flow-best-practices` folder within the `exchange` docset:  
`https://docs.microsoft.com/Exchange/mail-flow-best-practices/troubleshoot-mail-flow`
- A `guarded-fabric-shielded-vm` folder within a `security` folder in the `windows-server` docset:  
`https://docs.microsoft.com/windows-server/security/guarded-fabric-shielded-vm/guarded-fabric-deploying-hgs-overview`

Follow these rules for naming subfolders:

- Avoid product and technology abbreviations, which may not be meaningful to readers and are bad for [SEO](#).

- Avoid extra words that add length without meaning.
- Use **only** lowercase letters, numbers, and hyphens in subfolder names. Don't use spaces or punctuation characters.
- Avoid redundancy with the base URL. For example, if the base URL is `nuget`, avoid folder names like `nuget-get-started` or `nuget-samples`.

## Define a base URL

Each docset has both a base URL and a base path.

### Base paths

The path to which content is published for a docset is the base path. The base path starts with the first subdomain (the base URL) and can include subsequent sections of the subdomain. For example, `/azure` and `/azure/devops` are base paths for two separate docsets and share the same base URL (<https://docs.microsoft.com/azure/>).

You define the base path for a docset through the [OPS portal](#):

<b>Base Url</b> <a href="#">?</a>	<code>https://docs.microsoft.com/en-us/</code>	<code>platform-docs</code>
-----------------------------------	--	----------------------------

#### NOTE

We are aware that the terminology in our internal interfaces is inconsistent. The OPS portal, as pictured above, allows you to set the **base path**.

### Base URLs

The URL address that includes the first namespace (subdomain) after the domain of the site is the base URL. For example, <https://docs.microsoft.com/azure/> or <https://docs.microsoft.com/microsoft-365/>

An acceptable base URL:

- Uses standard ASCII alphanumeric characters A to Z
- Uses lowercase letters
- Uses American English
- When using multiple words, uses hyphens to separate them (not underscores). Do not concatenate the words.
- Uses intuitive language, preferably nouns
- Is meaningful about the intent of the content
  - Indicates a product, product family, category, or subject that's meaningful to the target audience
  - Is specific enough to be unambiguous but not so granular as to only cover a small number of pages
  - Is consistent with site architecture and existing base URLs
  - Does not duplicate or overlap with other base URLs or deeper subdomains
- Does not use acronyms unless using the full spelled out name would make the URL too long or complex
- Does not use code names
- Avoids using dates and numbers

New base URLs need to be approved by the DevRel Information Architecture team. They are evaluated against these criteria:

CRITERION	0: UNACCEPTABLE	1: SUBOPTIMAL EXCEPTION	2: MEETS STANDARDS
URL Hygiene	Concatenating words Underscores	Dates and numbers that have been approved by Marketing/CELA	Uses standard ASCII alphanumeric characters A to Z Uses lowercase letters Uses American English Uses hyphens between words, where spaces would normally be used
SEO	A project codename that hasn't gotten approval from marketing/CELA A base path that is on or synonymous with a term on the reserved word list A very generic term with no plan for extending it to more specific child nodes	A project codename that has gotten approval from Marketing/CELA A very generic term (e.g. /support) with a plan for extending it to more specific child nodes	Indicates a product, product family, category, or subject that's meaningful to the target audience Is specific enough to be unambiguous but not so granular as to only cover a small number of pages
Information architecture	Duplicates existing base paths (e.g. /microsoft-teams and /microsoftteams)		Does not duplicate or overlap with other base URLs or deeper subdomain (e.g. if /windows already exists, we don't want to add /windows-terminal as a base path. It should go under /windows/terminal. Then the owner of the /windows base path needs to approve it.)

The review process for base URL governance results in one of the following:

- Approval of requested base URL
- Recommendation of a different base URL
- Recommendation of a path under an existing base URL pending approval of that base URL's owner

#### NOTE

URLs must be consistent, hierarchical, and as shallow as possible to perform well in search. The more elements (folders) in the URL path, the harder for a search engine to crawl the content. Try to limit URLs to 2-3 levels deep from the base URL <https://docs.microsoft.com/{lang-locale}/>.

#### Create default URL

To avoid a 404 page when navigating to the parent directory of a URL, you can use an *index.ym*/ file in each folder. Here is what you need to do:

1. Create an *index.ym*/ file in the folder you would like to control.
2. Add content to this file. Keep in mind that it's the default file users see when they type only the parent directory.
3. Use this file as the href for the folder in *toc.ym*/, so when it displays, it shows the full TOC.

#### Remove folder from URL

We recommend that you follow the folder structure in your repo for your URL structure. However, in rare cases you might want to have a different repository structure than your URL structure. OPS provides a way to skip the folder name from your URL.

To skip the folder name, add a rule in the *build/content* section of the *docfx.json* file for your docset. For example, if you have a folder *foo* in your repository that you want to remove from the URL, add the following configuration:

```
{  
  "files": ["*.md"],  
  "src": "foo",  
  "dest": "."  
},
```

The logic is to "copy" the src folder *foo* to the dest folder, and keep the relative path inside the src folder as is. The above configuration specified the src folder while the dest folder is also specified in *docfx.json* as a required parameter.

For any folders you would like to strip out from the URL, nested or not, add a configuration for that folder.

**TIP**

If the files in the folder are not changed, the new paths may not appear after *docfx.json* is changed. To ensure that the new folder structure is applied, do a full build of the website branch from the [OPS portal](#).

## Share base URL

Multiple docsets can share the same URL, which is managed by the repo admin via the OPS Portal. For more information, see [Repo and docset setup](#).

# Troubleshoot SEO issues

4/16/2021 • 2 minutes to read

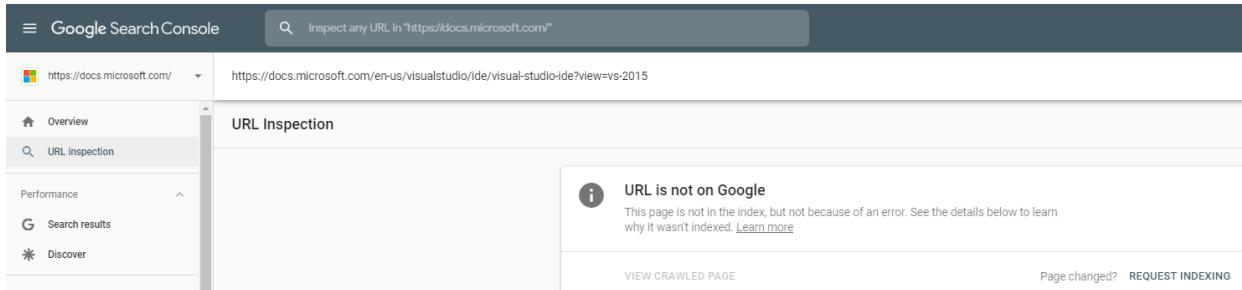
Having difficulties finding your content in Google, or wondering about indexing of the content in Google search? You can use some of the tips below to troubleshoot the issue.

## How to check content is in sitemap file

Go to [Docs sitemap file](#) and search for your content base path. For example, if the content is under "system center" you can search for it and then look for the URLs on the system center sitemap: [system center sitemap](#).

## How to verify the content is indexed in Google

After verifying that the content is in our sitemap file, you should go to the Google and type *Site:URL* and Google should show just that URL in result. Google search console, URL inspection tool also provides the indexing data.



The screenshot shows the Google Search Console interface. In the left sidebar, 'Overview' is selected. The main content area shows a URL inspection for the page <https://docs.microsoft.com/en-us/visualstudio/ide/visual-studio-ide?view=vs-2015>. A prominent message says 'URL is not on Google' with a note: 'This page is not in the index, but not because of an error. See the details below to learn why it wasn't indexed.' Below this, there are buttons for 'VIEW CRAWLED PAGE', 'Page changed?', and 'REQUEST INDEXING'.

## How to verify content is indexed in Bing

The Bing team has provided the tool that allows users to find whether content is indexed in Bing and served. In the Edge browser, go to <https://bingdex.binginternal.com/#/> and enter any Docs URL to find out if the page is Bing indexed or not.

## Meta description not appearing in Google or Bing

A lot of times writers find that Google or Bing does not show the meta Description as entered in content meta description in YAML section. Meta description is not used for ranking or relevancy but it helps with click through rate, which helps in relevancy. If Google finds relevant search phrases in meta descriptions then only, it will show otherwise many times it comes up with its own meta description.

## Need more help on SEO

If you need more SEO help to resolve content discoverability issues, send mail to SEO PM khairunj@microsoft.com to troubleshoot indexing issues.

# How to run an SEO hack to improve search discovery and content UX

11/2/2020 • 5 minutes to read

SEO hacks are a good way to focus writing efforts on making improvements to content that increase search traffic, while sharpening your team's web writing skills.

## How do you plan for an SEO hack?

Some up-front work is required before you're ready to run an SEO hack:

- **Check search performance:** Does your doc set have low search referrals (below 40% for a mature doc set)? Are there articles with overall high traffic, but low search traffic?
- **Request an SEO audit:** Check with [Khairun Jamal](#) about the availability of an SEO audit for your docs.
- **Scope the hack:** How much time do you have? How large is your content set, and how many writers do you have?
  - Focus on docs with the most traffic and/or strategic value.
  - Check docs with low overall traffic, too. They may be dragging down the doc set. Determine whether to edit them or remove them.

PAGE ELEMENT	SEO IMPACT
Page title	Highest
Heading 1	High
First para, meta description	Medium-high
Alt text and image filenames	Diagrams/Medium-high, Screenshots/Low
Subheadings (H2, H3)	Medium

## What happens the day of an SEO hack?

You're ready to proceed, when:

- You know the scope of your hack.
- You've determined what pages of content you want to cover.
- You have data and/or an SEO audit report guiding areas of focus.

SEO hacks are most successful and easiest to track when they include the following steps:

1. **Train on basic SEO:** Make sure content authors understand SEO writing techniques and what you'll work on in the hack. Use job aids, such as the [SEO cheat sheet for writers](#).
2. **Focus the effort:** Focus on the highest priority articles, text elements, and so on.
3. **Release simultaneously and measure results:** Release the edited content at the same time - or over just a few days. You can use a custom tag to filter on the content you hack. Then, measure the results of your hack.

## Check for these important issues, too

Some other issues are worth checking as you work through content:

- **Brand in page titles:** Microsoft Docs isn't a site for a specific product. So, it's important that each page title has the product/service brand or technology name. Data show that content draws significantly more search traffic when this important piece of information is in the page title. Compare:
  - **Set up a build environment | Microsoft Docs:** This title could refer to many different services, products, or technologies.
  - **Set up a build environment - ASP.NET | Microsoft Docs:** Adding the technology name provides necessary context. In one experiment, it increased search page clicks by 40%.
- **Filename relevance:** Changing and redirecting filenames that are irrelevant to content can help search performance. Article content can change significantly over time with multiple revisions. The filename is one of the signals for the content intent in the search result, and it contributes to search rank.

However, there's a lot of overhead to changing filenames. Make the judgment call:

- Leave the filename as is: Your filename isn't confusing and is relevant enough to the intent of the article.
- Change the filename and redirect: The filename is misleading or confusing. It's a complete mismatch with the content intent.

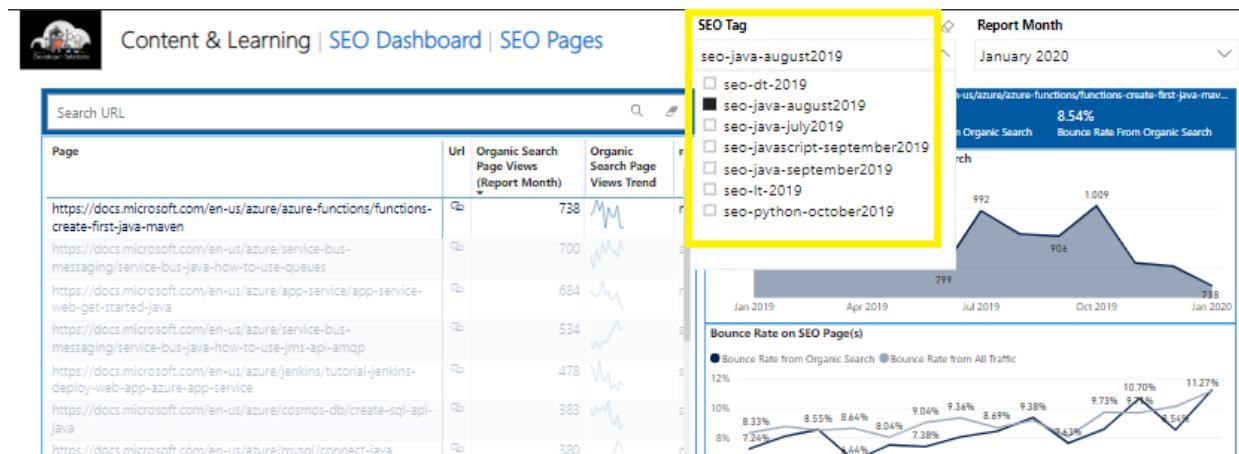
## Release docs and measure results

In order to see measurable impact, you need enough data. If you want to see the results for a content set, plan on releasing all changes simultaneously or at least within a few days of each other.

The search performance of a body of content is more than the sum of each part - its search performance is interconnected. An article that achieves higher rank can improve the performance of all articles in set.

### Optional: Custom tags

You can use custom tags to identify the content you hacked. Check impacts on metrics using the [SEO Dashboard](#).



### What to measure

You'll measure how the changes you make affect the content performance in search.

Pull data points from the [Content Performance dashboard](#):

- % referrals from search (% PVs from search): Of all traffic, the percentage from search. About 50% of all web traffic comes from search, and we know our customers often use Google to navigate to our content.
- Number of PVs from search: The number of PVs from search. The percentage of search referrals can be

impacted by other sources of traffic, such as conferences, campaigns, or links from prominent sources. You may see an increase in search traffic, even if it dips as a percentage of all traffic referrers.

- Bounce rate: Increase in bounce rate may signal a relevance issue.

Data points from [Google Webmaster Tools](#):

- Impressions: Number of times a webpage is listed on a search page.
- Clicks: Number of times a search result for a webpage is clicked.
- Click-through rate (CTR): The percentage of clicks a webpage receives on the search page relative to the number of times it appears in search results. Calculated from Impressions and Clicks.
- Average position: The average search rank of a webpage.

### Time periods to measure

Because you're essentially running a before/after content experiment, you need to compare the after period to a before baseline.

Time periods for before/after experiment data:

- **Before: 4 weeks (28 days)** of data before content was released.
- **Don't measure in-between time: 2 weeks (14 days).** After content is live, allow two weeks for content to be indexed and for changes to start impacting search performance.
- **After: 4 weeks (28 days)** after period allowed for search indexing. The further you get from the hack, the harder it is to assume causality between the hack and search performance.

Continue to monitor trends in your content following the hack. How people interact with your content in search affects its search performance: What terms people use to find your content; how often they click your content on the search page; and whether they stay on the page or contribute. Search performance is dynamic and always changing. You might continue to see improved results, your results might plateau, or you might see ups and downs.

## Related

- [SEO for writers: Basic techniques to improve search discovery](#)
- [How to write good page titles](#)
- [How to write good H1s](#)

# Content freshness definitions and maintenance checklist

6/16/2021 • 3 minutes to read

"Content freshness" refers to the time that has passed since someone fully reviewed an article to ensure the article is still technically correct, relevant, and complete. Content freshness is a key aspect of content maintenance.

As an article ages from its last freshness update, it goes from fresh to stale, and if it is not updated it eventually is considered abandoned from a content maintenance perspective.

ICON	DEFINITION	CONTENT PERFORMANCE DASH LISTING
	Most fresh (0-60 days)	A: 0-30, B: 31-60
	Fresh (61-90 days)	C: 61-90
	Stale (91-120 days)	D: 90-120
	Abandoned (121 days +)	E: > 120

You can find freshness data for your articles using the Content Performance Dash (<https://aka.ms/contentperformancedashboard>). Choose the Documentation tab at the bottom of the Power BI report page.

## Why does freshness matter?

The **ms.date** metadata value is published at the top of every C+AI technical document on docs.microsoft.com. Customer research indicates that customers use the date on an article as a proxy for determining if the content is relevant and trustworthy. This date is a contract you as the author make with the customer to let the customer know when the article last had a major review and update.

## How can you prioritize for the most impact?

You can use the content performance dashboard to find your highest traffic articles. You should plan a rotation so that your highest traffic articles are reviewed and updated frequently to ensure customers have a good experience on those high traffic articles. You should consider [archiving low traffic articles](#).

## How to mark an article as "fresh"

Content freshness reporting is driven by the **ms.date** value. When you do a full review of an article or make major updates, change the **ms.date** value to the date you made your edits. Don't update the **ms.date** value when you make typo fixes or minor, quick updates. Content freshness reviews are comprehensive reviews of the full article - it's your promise and contract with the customer that the content is up-to-date and accurate.

**NOTE**

`ms.date` is entered as a UTC date, with no time. 0:00 is therefore assumed. The date on the top of articles is shown in the user's local time. Given that, the date you enter and the date you see may be different.

## What should you review and update when you do a freshness pass?

WHAT TO CHECK	HOW TO FIND RELEVANT DATA
Review the entire article for accuracy, relevance, and completeness for the scope of the article.	
Review all screenshots, ensure that they reflect the current customer-facing UI.	
Check that all links go to current, accurate, and relevant content.	
Check verbatim comments for the article.	See the <a href="#">verbatim comment report</a> .
Check for GitHub issues for the article.	On the <b>Issues</b> tab of the public repo that contains the article, search for the file name - the feedback control inserts the file name into the issue comments, making this searchable item.
Check for open pull requests in the public repo.	See the <a href="#">pull request report</a> .

## Typical content dissatisfaction drivers

The table below shows the typical actions you can take to improve the customer satisfaction on a given article:

ARTICLE ISSUE	DESCRIPTION
Incomplete or inaccurate article	The article contains missing or incorrect information. Pulled from customer comments.
Needs example or bad example	The article contains a missing or incorrect example, which is normally reflected by customer feedback.
Broken links	The article is broken (404 error) or contains broken links.
Localization issue	The article is not translated correctly, or it is available in English only. Typically derived from customer comments.
Stale article	Articles with an <code>ms.date</code> value older than three months. If the article's date is not recent (stale), customers are less likely to trust the information.

# Docs.microsoft.com archive, retire, and removal policy

5/12/2021 • 3 minutes to read

## Purpose

The lifecycle of technical content on Docs.microsoft.com from creation to decommissioning aligns with the lifecycle of the business and customer need for the content. In most cases, transition from one phase to another follows Microsoft supportability guarantees, but should be flexible enough to handle outlier business cases as well.

This document describes the phase definitions and responsibilities required during each phase transition.

## Scope

This policy applies to all organizations who have published or intend to publish technical content on Docs.microsoft.com.

## Review and update cadence

This document will be reviewed and updated annually at the start of each fiscal year by the Docs.microsoft.com platform team.

## Definitions

- Current ecosystem:** Technical content that drives business value, is actively maintained. The content is often the most recent version or in canonical technical reference for a given product or technology.
- Archive:** Technical content that is near the end of its standard support lifecycle, is no longer driving sales or other business value, is maintained at the lowest priority level (sans P0/S0 security-focused updates) and other purposes defined by different content teams. Content in an archived state is moved from doc.microsoft.com to docs.microsoft.com/previous-versions with the Archive banner present.
- Archive banner:** As of Oct 2018, the banner states, "We're no longer updating this content regularly. Check the Microsoft Product Lifecycle for information about how this product, service, technology, or API is supported."
- Retirement:** Technical content that is out of standard support lifecycle and no longer driving sales or business value. Retired content will be moved from doc.microsoft.com to docs.microsoft.com/previous-versions with the Retired banner present.
- Retired banner:** As of Oct 2018, the banner states, "This content has been retired and may not be updated in the future. The product, service, or technology mentioned in this content is no longer supported."
- Removal:** Old technical content and content assets outside of standard support lifecycle will be removed from the website public view and search engine results. There are typically two patterns to retirement: single article and product/version. Content won't be accessible by customers.
- Compliance with Regulatory obligations and Microsoft Interoperability Principles:** Several compliance obligations and Microsoft Interoperability Principles require us to document and make publicly available protocols and APIs used to communicate with our High-Volume Products. More details are available at <https://msdn.microsoft.com/openspecifications/dn646764#Principles>

## Responsibilities

1. **Product/Technology GM:** The transition of technical content from one phase to another should be in alignment with the overall business objectives. The Product/Technical GM within the engineering team for that product or service is accountable for sign-off to transition content from active to archive/retire/removed state.
2. **Platform Team PM:** Collect the needed information from content team and justification as to why we want to move things to a different state.
  - Works with Anti-Trust council for approval to move state of content. Provide all information to Product/Technology GM and gets sign-off.
  - Work with content publishing manager to coordinate, follow-up on the phased content moves and maintain the status.
3. **Content Publishing Manager:** Content Publishing manager or delegate will coordinate the phased transition of technical content with the platform team PM. They ensure that the appropriate process is followed. They maintain the current status of content sets within their scope, with particular emphasis on the timing of state transitions in an internal public record.

While not a requirement, content teams are encouraged to coordinate community communication through Product/Technical Marketing to ensure transparency on the changes to technical content status. And to inform CSS / product support teams of the status of the content.
4. **Microsoft (Competition Law) Legal Counsel:** Will work with Content Publish Managers or delegate for technical content sets to meet compliance requirement and/or Microsoft Interoperability Principles. Windows XP, including the .NET Framework, Windows Server 2008, SQL Server 2008, Office 2007, Exchange 2007, Office SharePoint Server 2007, and future versions of these products will receive appropriate legal review to ensure that transition from one phase to another (Active -> Archive -> Retirement) meets legal requirements.

## Accidental publishing

The capability to remove published content from Docs.microsoft.com also enables content teams to rapidly unpublish technical content that was accidentally or inappropriately published. Such content can cause confusion or contain misleading message. In this scenario, the content publishing manager has the authority to revert the publication without #signoff from the Product/Technical GM, Microsoft (Compliance) Legal Counsel, and Product/Technical Marketing.

Content Publishing managers are required to minimally inform the Product/Technical GM of the action and log the change in the internal public record/report.

# How to retire, move, or rename a technical article

5/10/2021 • 4 minutes to read

This guidance is for *authors* of docs.microsoft.com technical articles that need to be deleted, moved, or renamed.

If you're a member of the docs.microsoft.com community and you think an author should retire an article for any reason, leave a comment or file an issue on the article to let the author know something is wrong with the article.

When authors want to retire, rename, or move articles, they need to follow specific steps to avoid bad experiences on the web site. The goal should be to gracefully retire content. Users of the website shouldn't find broken links or receive 404 errors.

## TIP

This article describes how to manually retire, move, or rename individual articles. If you need to move many articles, see [Moving or refactoring files in a repository](#).

## Step 1: Set the article to NOINDEX and republish it (as appropriate)

Do this step if you're preparing to deprecate content and don't want it to be discoverable, but do want it to remain published to support inbound links. To do so, add this line as the last entry in the metadata section of the article:

```
ROBOTS: NOINDEX
```

By using `NOINDEX` alone, you allow cross-links to current content to be crawled, and you avoid creating a dead-end for search crawlers.

## Step 2: Delete, move, or rename the file and create a redirect in the redirect file in your repo

To make sure that customers don't experience broken links, you need to create redirects for all articles that you're deleting, renaming, or moving. You create the redirects in the primary redirect file (`.openpublishing.redirection.json`) at the root of your repository.

- **Retire/delete:** Make sure you're listed as the author of the article in the `author` metadata attribute of the article. If you're not, update the `author` attribute, commit, and then delete the article. In the main redirect file, add the old article and the page to which that URL should redirect. The pull request for an article deletion must include the redirect as well.
- **Rename:** Create a copy of the article, give the file its new name. Then, delete the old article and add the redirect to the main redirect file.
- **Move:** Functionally, moving an article is the same as renaming an article. Create a copy of the article in the new location. Then, delete the old article and add the redirect to the main redirect file.

See [Redirect obsolete articles](#) for details on how to redirect articles.

## Step 3: Remove or update all cross links to the article from the

# technical content repository

Don't rely on redirects to take care of cross links from other articles. Update or remove the cross references to the article you're retiring, renaming, or moving. You can even do so for links in articles owned by other authors.

1. To make sure you're working in an up-to-date local branch, run `git pull upstream master` (or the appropriate variation on this command).
2. Scan the appropriate subdirectories in your repository for any articles and includes that link to the article you want to retire, move, or rename. For example, you might scan `azure-docs-pr/articles` folder and `azure-docs-pr/includes`. Either remove the cross links or replace them with an appropriate new cross link. You can use a search and replace utility to find the cross links if there's one installed. If you don't have one, you can search using Windows PowerShell for free! Here's how to use PowerShell to find the cross links:
  - a. Start Windows PowerShell.
  - b. At the PowerShell prompt, change into the related repository directory:

```
cd azure-docs-pr\articles
```

- c. Type this command, which will list all files that contain a reference to the article you're deleting:

```
Get-ChildItem -Recurse -Include *.md* | Select-String "<the name of the topic you are deleting>" | group path | select name
```

If you prefer to send the list of file names to a text file (in this case, named `psoutput.txt`), you can:

```
Get-ChildItem -Recurse -Include *.md* | Select-String "<the name of the topic you are deleting>" | group path | select name | Out-File C:\Users\<your account>\psoutput.txt
```

3. Add and commit all your changes, push them to your fork, and create a pull request to move your changes from your forked branch to the main branch of the main repository.

## Step 4: Publish

Publish your changes to the article repository by submitting a pull request. Test that the redirects work in staging before you sign off on the PR.

## Step 5: Cleanup tasks

Do these cleanup tasks immediately after the changes are published.

1. **Update the FWLink tool:** Check the [FWLink tool](#) for any FWLinks that might point to the retired/renamed/deleted article. [Update the FWLinks](#) to point to the appropriate replacement content. If you're not sure who owns an FWLink, or if you need to take over an abandoned link, review the [guidance on link ownership](#).
2. **Manage inbound links:** Determine if there are any high-traffic non-Microsoft inbound links to your content. Frequently, blogs, forums, and other web content point to articles. You can work with content owners to change or remove these links, and you can remove or update links from forum posts. Web analytics tools can tell you if there are any high-traffic inbound links you might need to manage in this way.
3. **Remove cached pages from search engines (only if necessary):** Do this ONLY if the content needs

to be removed from search quickly because of legal or severe customer issues. Per best practices from Google, normal priority content should be removed from search through natural search engine processes. Go to these web pages to remove cached web pages from search engines:

- [Bing](#)
- [Google](#)

4. **Clean up redirects:** The length of time an article-level redirect stays in place is TBD, based on your organization/service/product specific needs.

# Docs content lifecycle

6/10/2021 • 2 minutes to read

The goal of the docs.microsoft.com is to present the best content experience to our customers, to have the relevant fresh content present cleanly, to have a great search experience, and to avoid the clutter with outdated and stale content. This article describes the general guidelines for archiving the content from docs current site to the [docs previous version site](#)

## When to keep, archive, or cut

The archive strategy is across all content set, which is published from open publishing system on docs.microsoft.com site. Usually it's a decision of the content owner with product owner and marketing team to archive content but here are some of the general guidelines to follow when making archiving decision:

WHEN TO KEEP	WHEN TO ARCHIVE TO PREVIOUS-VERSION SITE	WHEN TO CUT

WHEN TO KEEP	WHEN TO ARCHIVE TO PREVIOUS-VERSION SITE	WHEN TO CUT
<p><b>Keep when:</b></p> <ul style="list-style-type: none"> <li>The page covers a product/service that is still supported <b>and</b> can be kept healthy and performant.</li> <li>Or, the page is legally required on our live site regardless of lifecycle, health, or performance.</li> </ul>	<p><b>Important!</b> Confirm with product engineering, marketing, customer support, and CELA that there is no legal requirements to keep this content.</p> <p><b>Archive if:</b></p> <ul style="list-style-type: none"> <li>The page covers a product/service that is OUT of the support lifecycle - OR- obsolete</li> <li>The page is for an on-prem product is on an extended support cycle and seldom maintained</li> <li>The page no longer maintained. (&gt;24 months since reviewed)</li> <li>The page has very low page views <a href="https://aka.ms/docs-low-no-pv">https://aka.ms/docs-low-no-pv</a></li> </ul> <p><b>Result:</b></p> <ul style="list-style-type: none"> <li>Page(s) moved to the previous versions site.</li> <li>GitHub issues and public PRs are turned off.</li> <li>Content marked as <code>NOINDEX, NOFOLLOW</code> for search engines. Only if archive content falls under high value product (HVP) category, can they be marked <code>INDEX, FOLLOW</code> : (* Product is under regulatory supervision) <ul style="list-style-type: none"> <li>Windows XP + greater (including .NET Framework) (*)</li> <li>Windows Server 2008 and forward (*)</li> <li>All versions of .NET Framework (not including .NET Core) (*)</li> <li>Office 2007 + greater</li> <li>SharePoint 2007 + greater</li> <li>Exchange 2007 + greater</li> <li>SQL Server 2008 + greater</li> </ul> </li> </ul>	<p><b>Important!</b> Confirm with product engineering, marketing, customer support, and CELA that there is no legal requirements to keep this content.</p> <p><b>Cut if:</b></p> <ul style="list-style-type: none"> <li>the page was published accidentally</li> <li>Or, if this content is legally required on docs.microsoft.com -and- the page is either <b>performing poorly</b> or offers no value to customers</li> </ul>

## Archive vs. retire

- Archive:** Technical content that is near the end of its standard support lifecycle, is no longer driving sales or other business value, is maintained at the lowest priority level (sans P0/S0 security-focused updates) and other purposes defined by different content teams. Content in an archived state is moved from doc.microsoft.com to docs.microsoft.com/previous-versions with the Archive banner present.
- Retirement:** Technical content that is out of standard support lifecycle and no longer driving sales or business value. Retired content will be moved from doc.microsoft.com to docs.microsoft.com/previous-versions with the Retired banner present.

When a product is out of support cycle, the related content should be archived or retired with a banner:

- Archive Banner:** We're no longer updating this content regularly. Check the [Microsoft Product Lifecycle](#) for information about how this product, service, technology, or API is supported.
- Retire Banner:** This content has been retired and may not be updated in the future. The product, service, or

technology mentioned in this content is no longer supported

# Archive and retire articles and folders by using the Docs Portal

6/11/2021 • 11 minutes to read

For any content management system and for content lifecycle, it's important to have an easy and dependable archiving workflow for successful migration of content. When product content is out of support cycle, or when a service is deprecated and its content needs to be archived, a content developer should be able to archive content from the [Microsoft Docs current versions site](#) to the [Microsoft Docs previous versions site](#).

The lifecycle of technical content on the Microsoft Docs site should also match the lifecycle of the product and services. Obsolete and retired product and services content should be retired and archived on a regular basis.

This article describes how writers can archive content in a few easy steps by using the Docs Portal.

## Get started

Before you archive any content:

- Confirm with product engineering, marketing, customer support, and CELA that there are no legal requirements to keep the content on the [Microsoft Docs current versions site](#).
- Confirm with the product or service team that content can be marked NOINDEX to remove it from search engines.
- Make sure to sign in to the Docs Portal to access the source repository, which is also known as a *repo*. If the current repo is a GitHub repo, make sure to sign in to the [Docs Portal](#) by using your GitHub account to check the access to the repo. If the current repo is the Azure DevOps repo, sign in by using your Microsoft account.
- Check with your documentation lead or M1 to determine whether the archive repo already exists. For example, if your content is in the `azure-docs-pr` repo, its equivalent archive repo, `azure-docs-archive-pr`, already exists. This repo should be used to archive content from the `azure-docs-pr` repo.
- Any contributor with read access to the repo should be able to archive content because the archive tool creates a fork. A pull request (PR) is submitted automatically from the fork.

## Supported archive and retire scenarios

With the self-serve archive tool, you can:

- Move a few articles from the current repo to the archive repo.
- Move an entire service or feature folder from the current repo to the archive repo.
- Archive a few articles by using a node from the table of contents (TOC).
- Archive conceptual content that uses a moniker and moniker range.

In the preceding scenarios, the self-serve archiving solution in the Docs Portal:

- Moves files physically from the current repo to the archive repo.
- Copies moniker or moniker range files of a specific version in an archive repo.
- Fixes all the links and images and moves the include files.
- Determines whether resources like images or include files are used in other repos by using built-in dependency-check logic. If these resources are used in other repos, the tool makes copies and moves the images and include files to the archive.
- Creates redirects for all the archived files and updates redirection.json files.

## NOTE

Currently, the automated archive tool doesn't support archiving Learn and Reference files.

For localized content archiving, see [Archive localized content](#).

## Archive tool

When you're ready to archive a part of a docset or an entire docset, you can use the archive tool in the Open Publishing System (OPS) documentation platform. When you archive content, it's moved from the current part of the [Microsoft Docs site](#) to the [Microsoft Docs previous versions site](#). Content in this archive site is considered no longer supported, but it might be needed by legacy customers for some period of time. Usually, the content is also removed from search engines to dissuade new users from trying to adopt deprecated content or capabilities.

At a high level, the steps are as follows:

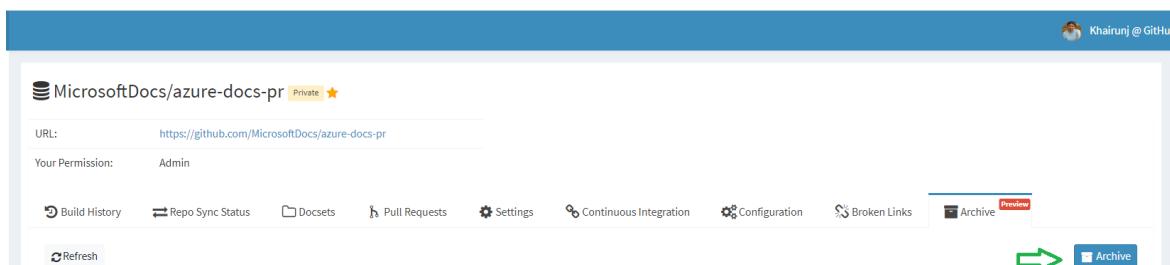
1. Confirm with your product partners that they're ready to archive content.
2. [Open OPS and the archive tool](#).
3. [Identify the source and archival destination target](#).
4. Run the tool to generate PRs in the originating repo and the destination repo.
5. In each PR, [review output, fix warnings, and sign off](#).

Let's get started.

### Open the archive tool in OPS

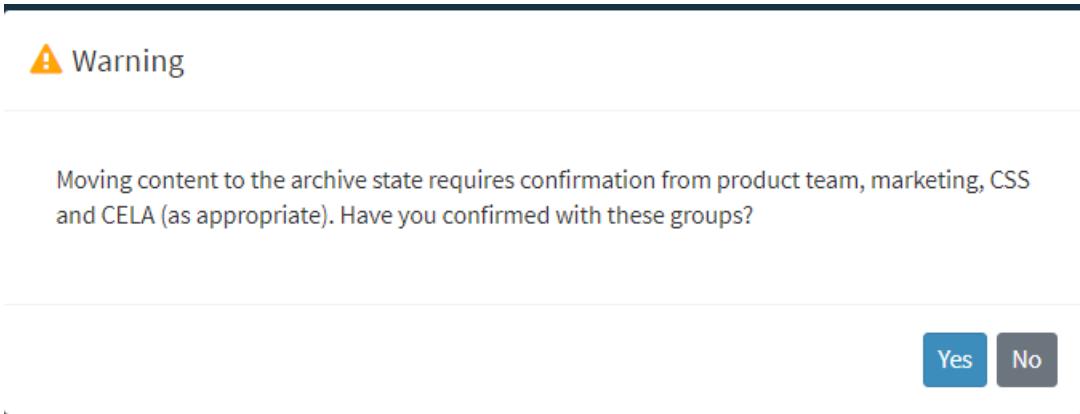
After you've gotten confirmation from your product partners, such as engineering, PMs, legal, and marketing, you can start the archival process.

1. Sign in to the [OPS Docs Portal](#):
  - Use your GitHub account if the content is currently in a GitHub repo.
  - Use your Azure DevOps ID if the current repo is in an Azure DevOps repo.
2. In the upper-left corner, enter the repo name that contains the content to be archived. For example, enter **azure-docs-pr**.
3. Select the name of the repo in the results.
4. On the repo page in OPS, select the **Archive** tab. The log of any previously processed archived jobs appears.
5. Select the blue **Archive** button to open the archive tool.

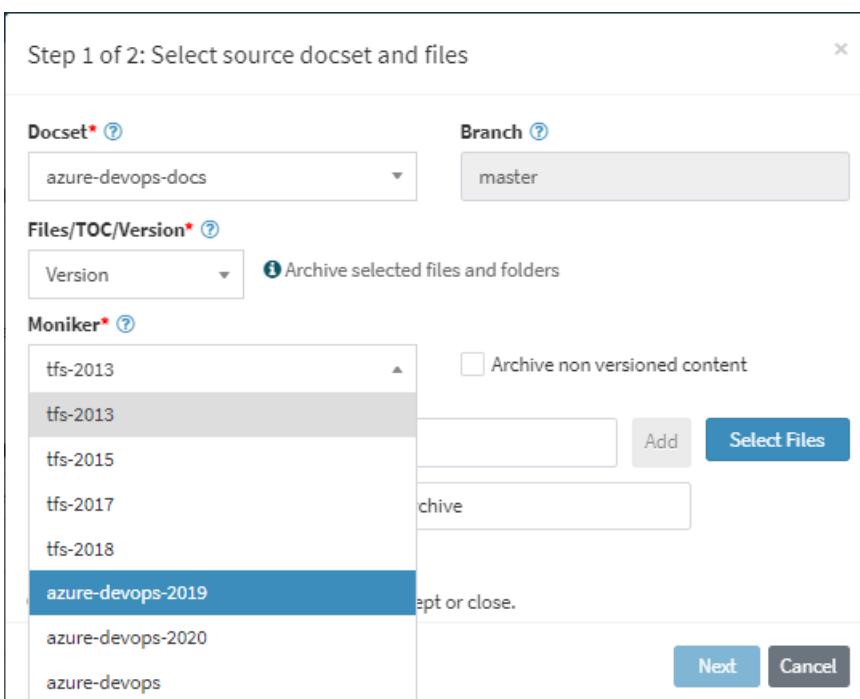


### Specify files to archive

- When the tool starts, select **Yes** to confirm you've checked that the product team is ready to archive.



- In the **Step 1 of 2** dialog box, specify the articles or version of the product to be archived.
- When you have both *versioned* and *nonversioned* content in the repo and you want to archive nonversioned content, first select the **Archive non versioned content** check box. Then select the button **Select Files**.



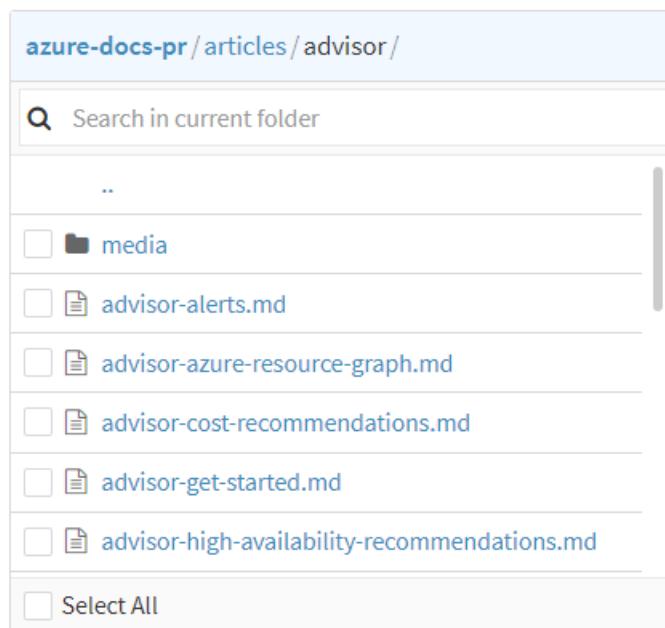
FIELDS	INSTRUCTIONS
Docset	Choose the docset name that contains the content to be archived. The "azure-docs-pr" repo has only one docset named <code>azure_documents</code> . Docset isn't a folder name in the repo. It's a grouping of content in an OPS platform. Usually, repos should have one docset only. But some repos have more than one docset.
Branch	This value is always <code>master</code> . This value can't be changed.

FIELDS	INSTRUCTIONS
Files or TOC	Specify the files to archive by using: - Files: Select files individually or an entire folder. - TOC: Archive all files referenced in a toc.yml file or a subset of a TOC.
Moniker	Select the version to archive. Then select files to choose the folder where the moniker or moniker range files exist.

#### NOTE

For version content with a single moniker reference in a file, the .md file is deleted from the source repo. For moniker range, the .md file is copied to the archive repo, and docfx.json is updated to only show the archive version at rendering time.

## File Selector



- After you select files or folders, select **Next** in the **Step 1 of 2** dialog box to start specifying the target destination for the archive.

### Configure the target archive destination

- In the **Step 2 of 2** dialog box, select the GitHub or DevOps account under which the destination repo you plan to use exists. This account is where the content will be archived.
  - If you signed in to OPS with your GitHub account, you see all the organization names of which you're a member.
  - If you signed in to OPS with your Azure DevOps ID, you see all the information for that account.
  - By default, **MicrosoftDocs** for the GitHub repo and **archive-docs** for the Azure DevOps repo will be selected.

## Step 2 of 2: Set target docset settings

**GitHub/DevOps Account\*** [?](#)  
MicrosoftDocs

**Repo Name\*** [?](#)  
azure-docs-archive-pr

Private [?](#)

**Docset\*** [?](#)  
Azure-docs-archive

**Branch\*** [?](#)  
master

**Docset Folder\*** [?](#)  
. . .

**Base Url: \*** [?](#)  
/previous-versions/azure

**Sub Folder** [?](#)

Preserve breadcrumb [?](#)

**GlobalMetadata**

NOINDEX [?](#)  NOFOLLOW [?](#)  is\_retired [?](#)

**Note:** Archive job will create a PR that you can accept or close.

[Previous](#) [Learn how to use archive tool](#) [?](#) [Archive](#) [Cancel](#)

- Specify the destination repo. If you know an archive repo already exists, enter that repo name in the **Repo Name** box. For example, much of the Azure archived content goes into the repo named **azure-docs-archive-pr**.

**IMPORTANT**

If the repo in which you want to create the archive doesn't yet exist, the tool creates the archive repo. For archive repo names, always use the same naming convention as the current repo and add **archive** in the repo name. For example, if the current repo is **azure-devops-docs-pr**, the archive repo should be **azure-devops-docs-archive-pr**. The repo name is autosuggested.

- Verify the values in the **Docset**, **Docset Folder**, **Base Url**, and **Branch** boxes. If you're using a newly created repo, make sure to use the term **archive** to clearly identify all archive docsets in the OPS platform.

- Optional configurations:

- Sub Folder:** It's fairly uncommon to use this option, but there might be use cases when you need it. To use a different folder structure than the current source repo, specify the new folder name in the **Sub Folder** box. For example, if a source repo has a folder named **active-directory** that holds the content being archived and you want that content in another folder, specify the name here.
- Preserve breadcrumb:** This check box should always be selected to preserve the breadcrumb in the archive repo. Only a few use cases would warrant removing breadcrumbs.
- GlobalMetadata:** Keep the default options unless you have specific needs. These metadata impact the following options:

OPTIONS	DESCRIPTIONS
NOINDEX	Provides crawlers with instructions for how to crawl or index web page content. Tells a search engine not to index a page. Because this content is archived, a life in search engines is usually undesirable. If the content needs to be discoverable in Google and Bing, turn off this toggle. Sometimes you might want a single page turned off. You should handle this scenario manually at an article level.
NOFOLLOW	Usually enabled for archived content. Tells a crawler not to follow any links on a page or pass along any link equity because this content is archived.
is_retired	<p>By default, all the archived content gets a generic banner. If the product team wants the content to have a retired banner text, turn on the <b>is_retired</b> toggle.</p> <p><b>Archive banner text:</b> "We're no longer updating this content regularly. Check the <a href="#">Microsoft Product Lifecycle</a> for information about how this product, service, technology, or API is supported."</p> <p><b>Retire banner text:</b> "This content has been retired and may not be updated in the future. The product, service, or technology mentioned in this content is no longer supported."</p>

5. Select **Archive** to run the job that generates two PRs:

- One PR removes content from the source repo and creates redirections.
- The other PR adds the content into the archive repo/docset.

#### Review the archive job summary and PR

1. After the archive job is successful, review the job status on the Docs Portal Archive tab. The **Summary** box provides information about the source repo and the list of files archived. If the archive job is for a TOC node, it also lists the TOC nodes.

The screenshot shows the Azure Docs Portal interface for an archive job. At the top, there are dropdown menus for 'Select a Docset:' (set to 'azure-documents') and 'Select a Status:' (set to 'succeeded'). Below this, a summary box provides details about the archive operation:

- Archived from docset:** azure-documents
- To docset:** Azure-docs-archive
- v-kents requested at:** 04/08/2020 10:40:46
- Status:** succeeded
- Execution time:** took 4 minutes

Below the summary box, there's a message: 'Please review the PRs and signoff: [PR in Source Repo](#) | [PR in Archive Repo](#) More details on Archive'.

Requested At:	04/08/2020 10:40:46	Started At:	04/08/2020 10:42:29
Completed At:	04/08/2020 10:46:51	Execution Time:	4 minutes
Detailed Log:	<a href="#">Details</a>		

**Summary:**

```

Archive Job Started
Configuration:
{
  "WorkingFolder": "D:\a\1\b\working",
  "ArchiveJob": {
    "JobId": "d7e2bac8-c995-46f8-8b3a-a76279e75a9c"
  },
  "Source": ...
}
  
```

2. Select each PR link, and review the changes created by the tool.
3. Open the PR for the source repo that was created by the tool. Review the proposed changes, but *do not enter #sign-off*.
4. Open the PR for the archive repo created by the tool.
  - a. Review the proposed changes. Preview the changes. Make sure the links, images, table of contents, and breadcrumbs are working.

b. If everything looks good, enter a comment in the PR:

```
#sign-off
```

c. If you created a new repo for this archive, you also have admin rights to merge the PR into `master` and into `live`.

5. *Wait until the archive PR changes are live.*

#### IMPORTANT

Make sure PRs are merged into `master` and `live` branches in the archive repo before you merge the source PR into master. Otherwise, the redirection won't work.

6. After the archive pages are live, go back to the PR for the source repo and add a comment in the PR:

```
#sign-off
```

## Update the archive landing page

The goal for this step is to make sure archived content is linked from the **Archive** landing page. For example, [Azure previous versions documentation](#) is a landing page for Azure archived content. To make sure there's a link on this landing page, create an Azure DevOps work item for the production team by using the information in [Hub and landing page updates](#).

## Archive YAML pages

In addition to Markdown files in a repo, YAML files are also found for other pages, such as hub and landing pages. During folder archiving, when an entire version or folder is archived, there are usually landing pages in YAML format, too. The archive tool supports the following types of YAML pages:

- Hub
- Landing
- FAQ
- Lifecycle
- Tutorial
- ZonePivotGroup

The archive tool copies `zone-pivot-group.yml` to the archive repo if it exists under the source repo root folder, regardless of whether the file is selected in step 1 or not. The archive tool won't handle the `zone-pivot-group.yml` file in other folders. The tool also doesn't respect if it's referenced in a `docfx.json` file.

When you use the archive tool, make sure to select the `.yml` files from the file list.

## File Selector

visualstudio-docs-pr / docs /

 Search in current folder

 xml-tools

 chromeless.yml

 default.yml 

 docfx.json

 index.yml 

 toc.yml

 vs-2015-archive.md

Select All

### IMPORTANT

Currently, no automated archiving solution is available for Learn and Reference content even though these files are also YAML-based files.

## Archive localized content

If the articles from a service that you're archiving have localized content, make sure to inform the [DevRel Global Experiences Loc Pillar PM](#) to manage the archiving of the localized content.

### IMPORTANT

Currently, if content is removed from a current repo, it's deleted from the localized repo automatically as part of the localization workflow. Not informing the localization team results in having no localized content on the archived site.

The localization team archives the localized content by creating the localized repo. They use the translation memory in most cases, and sometimes use their normal handoff or handback to archive the localized content.

## Submit the content archive request

If you have issues accessing the **Archive** button on the Docs Portal or you can't follow the preceding steps and have an urgent need to archive or retire content from the [Microsoft Docs current versions site](#) to the [Microsoft Docs previous versions site](#), you can submit the request by using the [archive request template](#) and the request will be handled by a content production team (CPS).

You'll be asked for the following information in the archive request template:

- The source repo and docset name.
- Either the file list, folder, or TOC node to archive.
- The target repo name if it already exists, for example, `azure-docs-archive-pr`, or the new repo name, which doesn't exist yet.
- Information for updating the archive landing page. Link the Label and the URL to point to it.

# Archive a docset

4/16/2021 • 3 minutes to read

The goal of the <https://docs.microsoft.com> is to present the relevant and fresh content experience to our customers. The lifecycle of technical content on docs site should also match the lifecycle of the product and services. It means that obsolete services and retired product content should be moved to archive site on a regularly basis.

This article describes the steps required to archive an entire **docset and repo**. This archive process will publish the content on docs previous version site [docs previous version site](#). If you want to archive just few articles from a folder or a TOC node see the information at [Archive and retire articles and folders using Docs portal](#).

## Summary of the required steps

When archive an entire docset, we do not move the physical location of the content, or change docset and repo name but instead make few edits in the configurations, and update the base path in [Docs Portal](#) and achieve the required end results.

Here are the list of steps that will be required to present the entire docsets content to previous version site.

- **Docset base path and redirection** – from "/xxx" to "/previous-versions/xxx"
- **Breadcrumb** – to add "Previous Versions" to breadcrumb
- **Metadata**
  - ROBOTS: NOINDEX,NOFOLLOW (to remove URLs from sitemap/search index)
  - is\_archived: true (to enable archive disclaimer)
  - is\_retired: true (to enable retire disclaimer)
- **Configuration:**
  - open\_to\_public\_contributors: false (to disable Edit button)
  - feedback\_system: None (to disable Feedback and GitHub issues section)
- **BI tracking**
  - docs\_archive: manual (to distinguish from archive tool in BI data)

## Detail step by step guide

### 1. Modify `docfx.json` and `.openpublishing.publish.config.json` on master branch

Following edits should be made in `docx.json` under global metadata section.

ROBOTS: NOINDEX,NOFOLLOW

is\_archived: true

"feedback\_system": "None"

"docs\_archive": "manual"

**TIP**

In some cases you might find Brand = Azure in global metadata section, make sure to remove that you correct previous-versions header can display.

```
"build": {  
    ...  
    "globalMetadata": {  
        ...  
        "ROBOTS": "NOINDEX,NOFOLLOW",  
        "is_archived": true,  
        "feedback_system": "None",  
        "docs_archive": "manual"  
  
    },  
}
```

Also make sure breadcrumb\_path in docfx.json points to previous-versions.

For example: "breadcrumb\_path": "/azure/cloud-solution-provider/bread/toc.json" is updated to "breadcrumb\_path": "/**previous-versions**/azure/cloud-solution-provider/bread/toc.json"

.openpublishing.publish.config.json Add config "open\_to\_public\_contributors": false

```
{  
    "docsets_to_publish": [  
        {  
            "docset_name": "archive-feature",  
            "open_to_public_contributors": false,  
        }  
    ],  
}
```

**2. Modify breadcrumb toc.yml on master branch**

a. Adding "Previous Versions" to breadcrumb, for example:

**Before**

```
- name: Docs  
  tocHref: /  
  topicHref: /  
  items:  
    - name: .NET  
      tocHref: /dotnet/  
      topicHref: /dotnet/index  
      items:  
        ...
```

**Example after adding previous versions in TOC.yml**

```

- name: Docs
  tocHref: /
  topicHref: /
  items:
    - name: Previous Versions
      tocHref: /previous-versions/
      topicHref: /previous-versions/
      items:
        - name: .NET
          tocHref: /previous-versions/dotnet/
          topicHref: /previous-versions/dotnet/index
          items:
            ...

```

#### NOTE

For breadcrumb to show previous-versions link make sure to add that on all tocHref and topicHref lines in toc.yml file.

### 3. Verify all changes except topic URL are ready on Docs review site

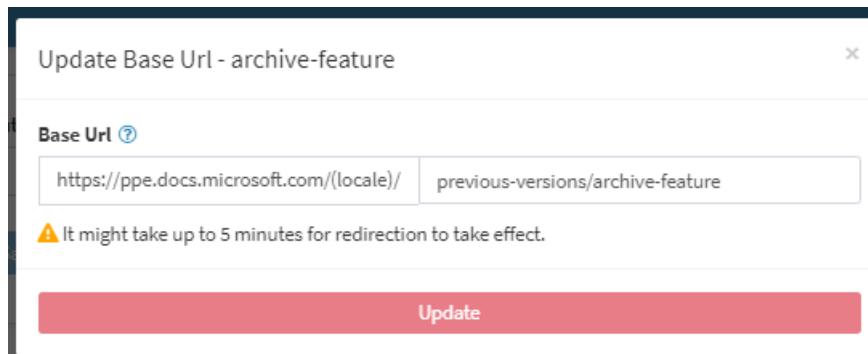
- Archive disclaimer is shown
- Breadcrumb is correct
- Feedback section and Edit button are disabled
- Topic URL is still the old one and not changed to /previous-versions

### 4. Change the base path from "/xxx" to "/previous-versions/xxx" on Docs Portal

Go to Docs Portal and find the docsets by using the search filter.

Go to docset properties and follow the instructions at

<https://review.docs.microsoft.com/help/onboard/admin/zero-downtime-changing-base-path?branch=master> to change the current base path. Here is an example:



This will automatically create a redirection rule from /xxx/\* to /previous-versions/xxx/\*.

### 5. Merge master branch to live

After all the above steps, the final step is to merge the changes from master to live branch.

#### NOTE

Keep few URLs from the current site handy to validate after the merge from master to live.

### 6. Verify on docs site (after live branch build is completed)

- New URL has "/previous-versions/xxx"
- Archive disclaimer is shown

- Breadcrumb is correct
- Feedback section and Edit button are disabled
- Old URL "/xxx" is redirected to "/previous-versions/xxx"

## 7. Modify docset Tenant to "Docs Archive" via Docs Portal

Go to Docs portal [docs portal](#) and view the docset settings. Go to Tenant field and change the value to Docs Archive.

**Base Url** [?](#)  
previous-versions/archive-feature  
Entire Url: [https://ppe.docs.microsoft.com/\(locale\)/previous-versions/archive-feature](https://ppe.docs.microsoft.com/(locale)/previous-versions/archive-feature)  
To change the Base Url, please go to 'Danger Zone'.

**Docset Folder** [?](#)  
docs  
To change this, please go to the repo and 1) change source folder name 2) change 'build\_source\_folder' in .openpublishing.publish.config.json file.

**Tenant** [?](#)  
 C+E  
 ASG  
 C+E  
**Docs Archive**   
 WDG  
 Test

'Tenant' is the team/organization your documents belong to. Choose 'Test' if this docset is for testing purposes.  
 Select All Locales  
 Please note that when you click Save, you will change the settings on all the locales of the docset, and the change will take effect on published pages after 2-3 minutes.

This does not change any content location but it is very helpful to track which docsets are archived.

## Archive a Repo

To archive an entire repo you will have to archive each docsets one by one by following the steps above.

After all the docsets are archived make the repo private in GitHub settings.

## Request to setup redirection to an absolute URI

If there is a need to setup redirections for all the articles on the docset to a specific URL submit the request <https://sitehelp.microsoft.com/new?fid=52> and engineering team will handle such request.

### Submit Request or Issue

The form below is used to report an LSI to request content updates such as Redirections, Archival, Retire, request Permissions, among other requests.

Where the change is needed or where the issue happens?

Select the service where you encountered the issue or request. In the case of an issue, think about the entire workflow instead of the individual tool or service category, i.e. localization, Publishing and OPS build are part of OPS Service

# Moving or refactoring files in a repository

5/21/2020 • 9 minutes to read

This article describes recommendations for moving files into a new structure in a repo in a way that makes sense for both writer workflow and review of the changes.

This article is overkill if you're just moving a few files - for that guidance, see [How to retire, move, or rename a technical article](#).

## Important things

First off, there are a few things you need to be aware of:

- **Don't move files using Windows Explorer:** The worst thing you can do is to use Windows Explorer to just drag and drop your files into new locations. Git cares about diffs, not files. Moving files by blunt force using Windows Explorer is likely to increase the amount of change in the diffs in your pull request. More diffs means the PR is larger and harder to review.
- **Limit your changes to folder names, file names, link fixes, breadcrumbs, and redirects:** Making other content updates completely changes the review requirements and will make your life and the PR reviewers' lives infinitely more complex and miserable. Stick to folder names, file names, link fixes, and redirects. Once they're all in place and merged to your release branch, then you can make other content changes. Sticking to these limits also makes large pull requests easier and faster to review for everybody involved, including you.

## Move and rename scenarios

There may be other scenarios, but these seem to be the core ones:

1. You need to rename a folder because your service name might have changed or because you need to rename a subfolder in your repository.
2. You need to split up articles, moving them out of one folder into two or more folders. This might include keeping the original folder, renaming it, or deleting it.
3. You need to rename a large number of Markdown files. This creates the same problems as moving files without renaming them because in Git, renaming and moving are fundamentally seen by the version control system as the same thing.

In all these scenarios, you have to think about:

- The actual moving and renaming of folders or files.
- Fixing all the links within the repository to the moved or renamed files.
- Creating redirects for all the moved files.

## Recommendations

### Write up your move scenario in detail and share it

Write up your move scenario in detail and share it with others to get feedback about the best way to move or rename your articles. Describe the current experience and the desired new customer experience.

### How can you make the task simpler? Can you use native Git functionality to minimize the apparent change?

Think about how to use Git so the changes appear as file and folder renaming instead of as new files.

For example, you might want to split your content into two new folders - all your articles are currently in folder1, and you want them to move to folder2 and folder3. You can potentially simplify your work in this case by renaming folder1 as folder2 as a first step. Then, move only the subset of files from folder2 into folder3. In this scenario, if you use the `git mv` command to change the folder name first, the pull request should clearly show that the files in folder1 were only renamed, with no other changes.

Example pull request: <https://github.com/MicrosoftDocs/azure-docs-pr/pull/38081/>

In this example, because the links in all the TOCs and articles were relative links, no broken links occurred when the folder was renamed. Only a few articles outside the folder had to be fixed manually.

However, if you use `git mv` to move/rename folders and files, redirects won't be created automatically. You'll have to:

- Manually create the redirects for those files.
- Or, request Akamai-level redirection if the number of files is too large. To request this, file a ticket using [Sitehelp](#), and select **Akamai redirection requests**.

### Scope your changes and use Git commands to minimize change

These recommendations are best applied to simpler scenarios and cases where there are 150 or fewer affected articles.

- **Consider using git-mv:** If you use Windows Explorer to move files, it's likely that Git won't recognize that the files in the new location are the same unchanged files that were in the old location. This makes the amount of change look large. But if you use `git mv` to move/rename files, Git recognizes automatically that the files are the same files, and the diff in Git and in GitHub will show that the file was renamed with no changes. This makes it possible for you to create a large PR that is easy to review. It also makes command-line diffing much easier.
- **Create multiple PRs that contain only one type of change:** Move files in one PR, then follow up with link fixes in a second PR. Add the redirects in a third PR. You can use the link validation warnings in the first PR as your working list for fixing the links in the second PR.

### Use Dr. Move strategically

Dr. Move is a tool to automate work related to renaming and moving files and folders. Dr. Move has some caveats and issues, but it does help you move lots of files into the right places, fix cross links, and automate the creation of redirects. After you run it, QA is imperative because the tool in its current iteration may fix cross-links that don't really need fixing. This happens because it can't distinguish between the overview.md file in the folder you are working with and the overview.md file in another unrelated service's folder. It "fixes" all the links to overview.md regardless of location in the repo. It also can't handle TOC files - they should be moved separately using `git mv`.

#### WARNING

Dr. Move is an unsupported tool with known bugs and limitations. It's incredibly useful, but requires very careful operation. Use it at your own risk.

For more information about Dr. Move, see the [README](#). If you can't see the README at this link, you must first join the [Azure GitHub organization](#).

### Pull requests and file moves

Follow these best practices around pull requests when you are moving and renaming folders and files:

- When you refactor or move content, make all the changes first to existing content. Don't mix new content with refactored content - keeping existing content separate from new content in a refactoring or move scenario makes it way easier to review and merge your pull requests easily so your project avoids delays.

- Contact the PR review team at [techdocprs](#) to let them know that a group of related pull requests are for your refactoring/move project. As long as you have broken the PRs up into manageable groups of changes, and the pull requests contain only moved/refactored existing content, they should be able to easily review the PRs.
- If you clearly communicate you have multiple PRs for a refactor or file move, and if the PRs are against a release branch, the PR review team is authorized to merge if there are broken link warnings due to changes being spread across multiple PRs. Other types of warning must be fixed prior to merge - the exception is only for broken links caused by the refactoring. After all the PRs are merged, the author should create a test PR to master to verify all the warnings are gone.

## Always use a release branch

Always request [a release branch](#) for moves and refactoring. A release branch allows iterative work and allows for a full review of the new staged experience for testing. Even if only one writer is making the changes, request and use a release branch.

After your release branch is created, make a local branch that tracks the upstream release branch:

```
git checkout -B <release-branch-name> upstream/<release-branch-name>
git pull upstream <release-branch-name>
```

Then, locally, create a working branch from the branch you created:

```
git checkout -B <working-branch>
```

This sets you up for local diffing during the QA stage described later in this article. Make your changes in this working branch.

## Always perform multiple QA passes locally and in the staged PR

The content author should perform the full QA described here.

### **WARNING**

Merging large pull requests without adequate review puts the repository at risk of unintended content deletions, unexpected formatting removal, regressions, and other problems.

### **Step 1: Create a pull request**

Create a pull request against your upstream release branch.

### **Step 2: Review the list of affected files with a local diff**

Everybody needs to know how to run a basic diff command locally!

Make sure the list of changed files includes only the files you intended to change.

These steps assume you followed the recommendation around branches provided earlier in this article.

1. In your current branch, make sure there are no pending changes:

```
git status
```

If you have pending changes, add and commit them.

2. Check out your local copy of the release branch:

```
git checkout <release-branch-name>
```

3. Update your local copy of the release branch:

```
git pull upstream <release-branch-name>
```

4. Check out the working branch that contains your move changes:

```
git checkout <working branch>
```

5. Run this command to compare the changes in your working branch to the content of the release branch:

```
git diff <release-branch-name> --name-status
```

6. Review the list of files in the output. Make a note of any unexpected files. Each file in the list will be marked as follows:

MARKER	DESCRIPTION
M	Modified
A	Added
R	Renamed
D	Deleted

### Step 3: Review the detailed local diff

1. Repeat the steps in the prior section, but modify the diff command to remove the `--name-only` parameter; run:

```
git diff <release-branch-name>
```

2. Page through the diffs using the space bar.

3. Investigate any unexpected files or changes.

4. To back out changes to an unexpected file, run:

```
git checkout <file path to file> upstream/<release-branch-name>
```

This restores the file to match what is in the upstream release branch.

### Step 4: Thoroughly review your pull request and the staged experience

1. In your pull request, review the highlighted diffs, and address any build warnings or errors. If there are broken links present because the PR is scoped to only the file move, make a note of that in the PR.
2. Verify the staged experience on the [review.docs.microsoft.com](https://review.docs.microsoft.com) site. The entire repo is staged with the changes in the pull request. You can access the staged content using the staging links in the PR or via the

<https://review.docs.microsoft.com> link for any article plus the PR branch, for example:

[https://review.docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal?  
branch=pr-en-us-9426](https://review.docs.microsoft.com/en-us/azure/virtual-machines/windows/quick-create-portal?branch=pr-en-us-9426)

Just replace the PR number with your PR number.

3. Type `#sign-off` in the pull request when you are sure the changes are all correct and you are ready for the changes to be merged. The pull request team will review and merge.

## PR review QA steps

To do the QA steps, first:

1. Create a local copy of the release branch and then pull the release branch content to your local computer.
2. Create a local working branch from the local release branch.
3. Pull the writer's pull request to your local working branch:

```
git pull upstream pull/<ID>/head:<branch>
```

Replace `<ID>` with the PR number. Replace `<branch>` with the branch listed on the right in the pull request - the one that's in the fork of the writer submitting the PR.

After you do this, you can run the diff commands described above. When you review, you're looking for:

- Are the changes clearly scoped to the content set being moved? If other people's files are affected, examine the detailed diffs closely to make sure they're related.
- Are there any systematic changes beyond folder renames, file names, link fixes, and redirects? If so, the PR shouldn't be merged.
- If there are lots of files appearing as new files, check carefully. This article provides guidance to help make sure most changes are seen by Git as file name changes only. File name changes don't normally appear in the GitHub diffs as green changes..

# Rename a product or service in content

5/25/2021 • 12 minutes to read

This article describes the approach we recommended for renaming a product or service in technical documentation. For example, this guidance applies when rebranding "Azure DocumentDB" to "Azure Cosmos DB". Some of the examples we provide are from that rebranding project.

Plan your updates when you first hear that a name change is planned. Even if the new name isn't final, start planning how you'll make the name change in the documentation:

- Phased approach or all at once? Do you have tight deadlines? Can you stagger the work? As you plan, consider your approach.
- Remember that in most cases, the name change can affect more than one repository for conceptual and reference content.
- Prepare to work proactively with your git repo admin to determine the best and safest way to merge your changes into the repo.

## Take an inventory

To plan appropriately, start by taking a full inventory of all relevant documentation assets. Doing this up front helps make sure you won't miss any content. With this approach, you can sequence, schedule, and delegate more effectively.

Make a list of the following assets. When possible, include the number of articles to update for each asset.

- **Git repo/s for conceptual content:** Count the number of articles and images to update. These articles are part of the core content for the product or service, including all the files your product/service owns. Also count all partner content created by other products or services that interoperate with your service. Partner content is stored outside of your core content folder.
- **Git repo/s for unmanaged reference content:** This REST content is commonly manually generated.
- **URLs for managed reference projects:** Projects such as .NET, Java, Node.js, Python, and other language reference content.
- **URLs for swagger-generated content:** Managed-generated content including REST and CLI content.
- **URLs for landing pages:** Landing pages that list to or link to your assets or reference your product name.
- **URLs for hub pages:** Hub pages that list or link to your assets or reference your product name.
- **URLs for dev center pages:** Dev center pages that list or link to your assets or reference your product name.
- **URLs for marketing or product owned web pages:** Web pages that need to be updated. Your partner team will likely handle the name change on their web page, but you'll need to provide an updated link to them. You may want to provide your partners with also known as links until your permanent new links have been created and published.
- **Links from the Azure portal.** Your partner team will handle the portal changes, but you'll need to provide updated links.

## Scope and organize rebranding work

Using your inventory:

1. Determine the scope of the work for each asset.
2. Organize and delegate the work.

Possible tasks:

- **Update naming in all articles:** Look for abbreviations and variations of the name.
- **Update file names:** File names that contain the product/service name may need to be changed. You can update the file name with the new product/service name or remove the product/service name from the file name altogether. You can remove the product or service name from the file name if it's already specified in the URL because of the git folder name.
- **Rename repo folder names:** Folders in a repo may need to be changed to use the new service slug, see [Create a new service slug](#). This task is non-trivial.
- **Create redirects:** Create a redirect entry for each file that was renamed or in a folder that was renamed.
- **Update TOC instances:** TOCs that contain either conceptual or reference content for your service may be managed outside of your normal working folders. Work with the [docs.microsoft.com](#) team to update all TOCs that you don't manage yourself. This task includes both conceptual, managed reference, and unmanaged reference content.
- **Update images:** All images should be reviewed and updated with the new product/service name.
- **Update embedded videos:** All embedded videos should be reviewed and updated with the new product/service name.
- **Updated the searchScope value:** In the `docfx.json` file for your repo so that the new name appears in the scoped search. The name you define in the searchScope appears in the search box at the top of [docs.microsoft.com](#) for some, but not all repos. It helps users find articles related to your product on docs. Use the full product or service name. The name appears in the metadata for all articles that should appear in the scoped search results. This name isn't required for Azure services in the `azure-docs-pr` repo. For more information, see [Search Scopes](#) and the [Site Search Vision Document](#).

If there are other tasks that need to be completed that are specific to your project, add those to your plan.

Once you've made your task list for each asset, organize the work. If you're rebranding in phases, we recommend this order:

1. Replace naming in each article in your conceptual repo and in landing and hub pages.
2. Update file names, folder names, URLs, and redirects to address renaming.
3. Work with your product team to organize the tasks for remaining assets.

## Create a new service slug

Once the new name has been finalized, work on creating a service slug. If the new name hasn't been finalized yet, you can continue with the rest of this article. Though, be sure to revisit this task later.

The service slug is part of the [docs.microsoft.com](#) URL. For example, the service slug for Azure Cosmos DB is `cosmos-db`, and the URL for the technical documentation is <https://docs.microsoft.com/azure/cosmos-db/>. If your content is in the `azure-docs-pr` repository, the service slug is also the name of your content folder. Your service slug should likely be the same slug that is used in the marketing/product documentation. For example, the service landing page (marketing material) for Azure Cosmos DB is <https://azure.microsoft.com/services/cosmos-db/>. As such, it shares the same service slug as the documentation.

Work with your product team and marketing to standardize on a new slug and use it consistently on all Microsoft assets.

Ensure that all partners are using the same hyphenation, for example `cosmos-db` and not `cosmosdb`.

## Work with your repo admin

First, complete your inventory and identify the work to do. Then, connect with your repo admin and discuss the

following concepts:

- **Content reorganization:** Discuss any overarching changes to the organization of the content. As products and services grow, some move from a single folder to multiple folders in git, and some centralize content from multiple folders into a single folder - or some other combination of movement. If you know of any reorganization efforts, let your repo admin know.
- **Batching work into small pull requests:** Discuss how you can complete your work while making pull requests that contain no more than 100 files. This planning ensures your PRs can be validated and merged into the system without introducing errors. Read [Moving or refactoring files in a repository](#) for guidance on creating pull requests. The article provides information for creating pull requests for large numbers of files before meeting with your repo admin.
- **Using a release branch:** Determine whether your renaming project should use a release branch to group all the updates together. If your renaming aligns with a conference, the changes will likely need a conference-specific release branch. Remember that release branches for conferences lock down at least a few days before the conference.

## Get production help

At some point in the process, you may want to automate finding instances of your product or service name. Provide the production team with a list of repos to search. They can then automate finding any remaining instances and provide an Excel workbook with the hits.

To get help from production, complete the following tasks:

- **Create a list of repos to search:** You can find a list of all ~2790 docs repos [here](#). The filtered list of 57 repos searched for the Azure Cosmos DB renaming is [here](#).
- **Open a production bug to search repos:** In the production bug, specify the list of repos to search and identify how you want the output presented. As an example, see [User Story 1208424](#). The user story shows how the production team searched 57 repos for remaining instances of "DocumentDB". Even after the cleanup, they still found hundreds of hits in 14 repos.
- **Review the Excel output:** Once production has completed the bug, they'll provide you an Excel workbook with the output. In the sample provided, the output was an Excel workbook in which a new worksheet was created each repo with search hits. Each worksheet contained the file name, the line number, and the line text. You can run this process multiple times. It takes hours to complete as the production person has to get permissions to access each repo, clone each repo locally, and then run the script.

If you want to attempt to run the automated script yourself, the Python script used by the production team is attached to [User Story 1208424](#).

Be sure that you update all product name instances and links. Remember links don't rely on the redirection file.

## Standardize naming

Once the new name has been finalized with the product group and marketing, create guidance on how the new name should be used.

Use the following questions to create guidance:

- Does Microsoft or Azure need to be included in every mention? Work with your marketing contacts to find the answer. Some services are Azure branded services and require Azure on every use, except for Service slugs and some TOC mentions.
- Are there any approved acronyms for the service?
- Are there any approved shortened versions of the product name?
- How does the new name need to be used in the Title meta data?

Work with Monica Rush to add the new guidance to the appropriate style guides. Here's an example of what Monica needs:

- **New name:** Azure Cosmos DB
- **Short reference:** Azure Cosmos DB
- **Date name change goes public:** XXX
- **Definition:** Globally distributed, multimodel database service for managing data at a global scale.
- **Notes:**
  - Always include Azure with Cosmos DB.
  - It's Cosmos DB, not CosmosDB.

## Create work items

When you know what needs to be done, create a single Azure Boards Feature for the renaming project. Then, create a child User Story for each work item that you identify in your inventory.

- One user story for all on-page updates in a repo
- One user story for all file name changes in a repo
- One user story for creating redirects
- One user story for all image updates in a repo
- One user story per reference repo to update
- One user story for all reference repo TOCs to update
- And so on

Assign these updates to Den Delimarschi (DOCS) (dendeli@microsoft.com).

There are many moving parts when you rename a single service. When you track each one independently, you can assign them to different individuals and close them as they're completed. Tracking the parts independently makes it much easier to manage.

For reference docsets, is your URL changing? TOC and URL changes are handled separately from each other. Create a work item for the TOC update and create a separate work item for the URL change.

Are you going to fix instances of the product/service name in other people's content? Are you going to ask them to fix it? Create work items in the correct Azure Boards project and area path for that team and assign them to individuals as appropriate.

Create a work item for each of the following items:

- **Update breadcrumb files:** For Azure services, update the breadcrumb file (articles/bread/toc.yml in azure-docs-pr) to use the new product name and folder location.
- **Capture search traffic to the old name:** Until the rebranding is widely known or customers have upgraded, they'll still search google with the old name. Consider adding a new article to your Overview section that describes the reason behind the name change. Use both names in the title to capture the search traffic for the old name. For example, an article was added to the Overview section titled "Azure DocumentDB users, welcome to Azure Cosmos DB" after the rebrand for Azure Cosmos DB. [Here's another example for R Server](#). An article like this helps users understand the transition and redirects search traffic.
- **Set up reporting:** Work with Ekaterina Lazhintseva (katyal@MICROSOFT.com) to ensure that data about the new content appears in the correct reports.

You may uncover instances of your old product name in the code or UI of other services. If you do uncover these instances, the docs can't be updated until the UI or code has changed. Create a work item and keep it open until the doc issue has been fixed. Then, connect your product group to the partner product group to make them aware of the issues you find. You'll likely find instances in your doc search that the product team doesn't know

about.

## Work with your product group

Work with your product group on the following tasks:

- **Update reference content:** Product teams need to update code comments for any autogenerated reference content, such as .NET reference pages or Swagger-generated REST content. Remind them of this task when they ask for doc changes and ask about deadlines. They may consider these “docs” and forget that they own them. Handle updates to the TOCs that point to reference content separately from the code comments. Contact the [docs.microsoft.com](#) team to make the TOC updates for reference content.
- **Maintain a list of external dependencies:** While taking inventory or making doc updates, you may find product name or service name instances in the UI or code base of other products and services. Relay these findings to your product group. They can work with the partner team to make the change and track the related doc updates. The doc updates can be made after the UI or code has been updated.
- **Identify who will update samples/code/templates referenced in the docs:** While reviewing your inventory and making updates, look for other resources that work with documentation. Use the resources to determine who should make the updates. For example, are there readme files for Resource Manager templates that need to be updated? Are there git samples cloned or downloaded as part of a tutorial? If so, who will update those git samples?

## Make the changes

Once you've reviewed your content set and come up with a plan with your repo admins, it's time to start making updates.

Follow the instructions in [Moving or refactoring files in a repository](#) to file pull requests for large numbers of files. Update your work items as you go.

## Verify your work

At any point in the process, open a production bug to have multiple repos searched for instances of the old product or service name. You may want to run this process after you think you've cleaned up all instances. Run the process at this time to ensure you've fixed everything.

# How to redirect obsolete articles

6/16/2021 • 4 minutes to read

To avoid broken links, redirect files that are removed, renamed, or moved to a new location. You redirect files by using a main redirect file in the repo. Or, you can redirect a file by requesting site-level redirection.

## Main redirect file

You can manage redirects in a single file at the root of your repo, for example `.openpublishing.redirection.json`. Using a single file makes it easier to track the files that have been redirected, and lets you remove the redirected files from the repo.

To add a redirect to the `.openpublishing.redirection.json` file, add an entry to the `redirections` array as shown in the following example:

```
{  
  "redirections": [  
    {  
      "source_path": "openpublishing/docs/partnerdocs/index.md",  
      "redirect_url": "/opsdocs/partnerdocs/VSTSgit-github",  
      "redirect_document_id": true  
    },  
  ],  
}
```

- The `source_path` is the relative link to the file you want to redirect.
- The `redirect_url` is the link to the target page. `redirect_url` can be a relative link to another file in the same docset, an absolute link to another article on `docs.microsoft.com`, or a URL to any web page.
- `redirect_document_id` indicates whether you would like to transfer the document ID from the `source_path` URL to the `redirect_url`. Use `true` for this setting if you want to preserve the `ms.documentid` attribute value from the redirected article, which will transfer reporting data such as page views and rankings to the target article. If `redirect_document_id` isn't declared explicitly, the system defaults to `false`.

### IMPORTANT

Using this method, you can preserve the document ID only if the target file is in the same docset as the redirected file. If you redirect multiple files to the same target file, you can only set `redirect_document_id` to `true` for one of them. If you are moving existing articles from one repository to another repository, you need to file a ticket with the OPS development team. They can assist with a programmatic solution that allows you to preserve the document ID.

After you've redirected a file via the main redirect file, remove the original file from the repo. Otherwise, you'll get a build warning like the following example:

```
[Warning] 'path-to-file.md' would be overwritten by redirection rule configured in main redirection file .openpublishing.redirection.json. Please remove the original file to resolve this warning.
```

For interactive tutorials and Microsoft Learn content, you can redirect from the `index.yml` file by replacing `index.md` in the main redirection file.

When a writer uses the automated archive tool to archive versioned content, it adds a monikers-related property in the main redirection file `.openpublishing.redirection.json`, for example `vs-2015`:

```
{  
    "source_path": "docs/vs-2015/azure/vs-azure-tools-access-private-azure-clouds-with-visual-studio.md",  
    "redirect_url": "/previous-versions/visualstudio/visual-studio-2015/azure/vs-azure-tools-access-  
private-azure-clouds-with-visual-studio",  
    "redirect_document_id": false,  
    "monikers": [  
        "vs-2015"  
    ]  
},
```

## How to generate a main redirect file based on file-level redirects

Before main redirect files were supported, we had to redirect files individually by adding the `redirect_url` and `redirect_document_id` (optional) metadata to the YML header of the Markdown file:

```
---  
redirect_url: https://marketplace.visualstudio.com/items?itemName=docsmsft.docs-authoring-pack  
redirect_document_id: false  
title: Gauntlet VS Code extension  
author: meganbradley  
ms.author: mbradley  
ms.date: 05/18/2018  
ms.topic: contributor-guide  
---
```

This approach is no longer supported. If you have a repo that still uses file-level redirection, you can clean it up by using the Docs Markdown VS Code extension functionality. This feature generates a main redirect file and removes the obsolete files from the repo.

1. Install the Docs Markdown VS Code extension, either [individually](#) or as part of the [Docs Authoring Pack](#).
2. Open your cloned repo and make sure it's up to date.
3. Hit F1 to open the VS Code command palette.
4. Start typing "Docs: Generate main redirect file" until you see the command, then select it.
5. The main redirection script runs on the repo. For each Markdown file it finds with the `redirect_url` attribute, it will:
  - a. Add an entry for the file to `.openpublishing.redirection.json`. The path of the file becomes the value of `source_path`. The value of `redirect_url` in the main redirect file is the same as in the source file.
  - b. If you included the `redirect_document_id` attribute in your Markdown file with the value `true`, it adds the `redirect_document_id` to the main redirect file. Otherwise, it defaults to `redirect_document_id equals false`.
  - c. Move the redirected file to a folder under your default path, such as `C:\Users\your-alias\Docs Authoring\Redirects\repo-name_deleted_redirects_date-and-time`.
6. When the script completes, check the deleted redirects folder to make sure the files were removed as expected.
7. If so, submit a PR to update the repo.

You can run the script multiple times if new file-level redirects creep into the repo.

After a few runs of the script, your `Redirects` folder might look something like this:

A screenshot of a file explorer window titled "Megan Bradley > Docs Authoring > Redirects". The list is sorted by name and shows the following entries:

Name
docs-help-pr_deleted_redirects_2018-6-19_17-36-658
openpublishing-docs_deleted_redirects_2018-6-19_17-54-487
openpublishing-docs_deleted_redirects_2018-6-19_18-47-266
openpublishing-docs_deleted_redirects_2018-6-20_16-53-530
openpublishing-docs_deleted_redirects_2018-6-20_16-58-760
openpublishing-docs_deleted_redirects_2018-6-20_17-16-608
openpublishing-docs_deleted_redirects_2018-6-20_17-48-965

## Site-level redirection

Site-Level redirection works for large content sets to move in bulk, but is available **only** for certain scenarios. If your redirection meets these requirements, you can request bulk redirection:

- Must be an MSDN and MTPS to Docs redirection scenario.
- The docsets must display different **Base Path** values in the Open Publishing Portal.
- Must be a custom redirection scenario that isn't achievable in a main redirection file.

If your request meets these requirements, file a site-Level redirection request by following these steps:

1. Go to <https://sitehelp.microsoft.com>.
2. Select **Submit request or issue**.
3. Select **Redirections** for the service, then the appropriate service category.
4. Follow the instructions provided.

# Use Multiple Redirection Files

3/5/2021 • 8 minutes to read

Historically, the single redirection file has had the following name: `.openpublishing.redirection.json`. The file has been located in the same root directory of the repo where the file `.openpublishing.publish.config.json` is located.

When a large repo has only a single redirection file, it can become oversized with a lot of redirects (e.g., Azure and Archive repos), which is difficult for content teams to maintain or update. We want to allow using multiple redirect files within a repo, to reduce the size of the single redirect file, and to make managing/updating redirects easier and more efficient for all users. Furthermore, there can be many occasions when two pull requests (PRs) suffer conflicts with each other. The enhancement that allows a repository to have multiple smaller redirection files can reduce the conflicts problem.

## New flexibility of using multiple redirection files

Starting January 2021, a repo is no longer limited to at most one redirection file. The following flexibilities are available:

- **File names:** You can add several redirection files. Each file name can vary within a restricted pattern, to maintain uniqueness and to be self-descriptive. Therefore several redirection files can be in the same directory. All redirection file names must obey one of the following two patterns:
  - `.openpublishing.redirection.json` (The historical single file name.)
  - `.openpublishing.redirection.<AREA_NAME_HERE>.json` (The added portion `.<AREA_NAME_HERE>` adds exactly one dot-delimited node to the file name. The added node must appear immediately before the `.json` extension. A plausible example name: `.openpublishing.redirection.virtual-machines.json`.)
- **Directory locations:** You can place redirection files in any directory.

## Syntax inside a redirection file

Inside a redirection file, each entry must provide the directory path in GitHub to the old *source* file. Historically the path had to be a *relative* path, in relation to the location of the redirection file. But now you can choose between using *relative* or *absolute* paths.

The `"redirect_url"` attribute is a partial https address, usually related to the Docs website. Therefore, between these source path examples, the `"redirect_url"` value is unaffected and does not vary.

**Relative path syntax:** `"source_path"`

The following JSON examples each use a relative path to the source file. Each code example starts from the very top of the redirection file.

### Example 1 of relative paths

- *Location of redirection file:* The repo root directory. An `articles/` directory also exists in the root directory.

```
{
  "redirections": [
    {
      "source_path": "articles/network-watcher/how-to-connection.md",
      "redirect_url": "/azure/network-watcher/tutorial-connection",
      "redirect_document_id": false
    },
    {
      "source_path": "articles/foo/aa.md",
      "redirect_url": "/azure/bar/bb.md",
      "redirect_document_id": false
    }
  ]
}
```

Note the following requirements in the preceding JSON example:

- The entry uses the `"source_path"` attribute, not `"source_path_from_root"`.
- The relative path value of `"source_path"` cannot start with a `/` character.

#### **Example 2 of relative paths**

- *Location of redirection file:* Under `articles/network-watcher/`, which itself is under the repo's root directory.

```
{
  "redirections": [
    {
      "source_path": "how-to-connection.md",
      "redirect_url": "/azure/network-watcher/tutorial-connection",
      "redirect_document_id": false
    }
  ]
}
```

Note the following requirement in the preceding JSON example:

- The `"source_path"` value must not include `articles/network-watcher/` because the redirection file is located in the `network-watcher/` directory.

#### **Absolute path syntax: `"source_path_from_root"`**

The option to use absolute paths was added in the same release that added the multiple redirection feature. Absolute paths seem like the better option. Absolute paths enable you to do the following things, without you having to edit or update any entries:

- Move a redirection file to any other directory.
- Recombine entries from two small redirection files into one file.

#### **Example 1 of absolute paths**

The following JSON example for absolute source paths starts from the very top of the redirection file.

- *Location of redirection file:* The repo root directory. An `articles/` directory also exists in the root directory.

```
"redirections": [
  {
    "source_path_from_root": "/articles/network-watcher/how-to-connection.md",
    "redirect_url": "/azure/network-watcher/tutorial-connection",
    "redirect_document_id": false
  }
]
```

Note the following requirements in the preceding JSON example:

- The entry uses the `"source_path_from_root"` attribute, not `"source_path"`.
- The absolute path value of `"source_path_from_root"` must start with a `/` character. In a sense, `"source_path_from_root"` starts from the repo root directory.

## Registering redirection files in repo configuration

Any additional redirection files must be listed (registered) in the `"redirection_files"` section in `.openpublishing.publish.config.json`. Redirection files that are not registered will be ignored.

The lone exception is that the historical single redirection file, located in the repo root directory, whose name need not be registered. This redirection file is implicitly registered, if it exists. It might be good to explicitly register this redirection file, for consistency and completeness.

### Full syntax of registration

The following JSON code snippet starts from the very top of `.openpublishing.publish.config.json`. The snippet shows the nesting level of the `"redirection_files"` array section.

```
{  
    "build_entry_point": "docs",  
    "redirection_files": [  
        ".openpublishing.redirection.json",  
        ".openpublishing.redirection.virtual-machines.json"  
    ]  
}
```

### Example registrations

Next are some examples of redirection files being registered in `.openpublishing.publish.config.json`.

#### Example 1 registrations

- Here all redirection files are stored together in the repo root directory.
- The writer felt like explicitly registering the historical file.
- All other redirection file names contain an extra node to describe their area, which keeps each name unique.

This might be the best design of the examples in this section. This design makes it easy for writers to find and see and access all the redirection files. It is no problem to have a long and growing list of these files in the root directory.

```
"redirection_files": [  
    ".openpublishing.redirection.json",  
    ".openpublishing.redirection.active-directory.json",  
    ".openpublishing.redirection.cognitive-services.json"  
]
```

#### Example 2 registrations

- Here all redirection files are stored separately, each in a lower level directory corresponding to the area covered by the redirection file.
- The redirection file names contain a helpful extra node to describe their area. The extra nodes are not needed for uniqueness in this particular scenario.
- The writer either chose to leave the historical redirection file to be implicitly registered, or there no longer is a file by the historical name in the root directory.

```
" redirection_files": [
    "articles/active-directory/.openpublishing.redirection.active-directory.json",
    "articles/cognitive-services/.openpublishing.redirection.cognitive-services.json"
]
```

### Example 3 registrations

- Here all redirection files have the same traditional name, but are located in different directories.

```
" redirection_files": [
    "articles/active-directory/.openpublishing.redirection.json",
    "articles/cognitive-services/.openpublishing.redirection.json"
]
```

## Switching from single to multiple files

When switching to multiple redirection files, first secure a consensus from your writing team. The single large redirection will be subdivided into multiple files. Announce a date.

Second, scour the single redirection file for errors to fix in a narrow PR. This will spare you of hassles later in the process. Fun fact is that most large redirection files have several errors. One common error is to have

`"redirect_url"` values that give GitHub paths instead of Docs Https paths.

Finally, consider using the following set of recommended steps the first time your repo switches to multiple redirection files.

### 1. Convert to absolute paths

Before you split the single large `.openpublishing.redirection.json` file into multiple smaller files, convert all entries from relative paths to absolute paths. These path conversions can probably be accomplished easily by the following one time change-all:

- *Find All:* `"source_path": "`
- *Replace All with:* `"source_path_from_root": "/"`

### 2. Sort the JSON entries

Suppose your plan is to split your one large redirection file into several smaller files. One smaller file will contain all the entries that involve Active Directory articles. Another will contain the entries for Virtual Machines, and so on. To accomplish these groupings of entries, you first need to sort the entries, perhaps by their `"redirect_url"` values. But how can you sort multi-line entries?

One way to sort JSON entries is to use the extension **Sort JSON Array** in the VS Code editor. But if the extension does not work as expected, another more laborious approach is described by the following steps:

1. Convert the JSON file into a CSV file.
  - This conversion flattens each multi-line entry into one row.
  - Website tools such as <https://CsvToJson.com/> can accomplish this conversion instantly and for free.
    - The Norton 360 security website says their checks have detected no risk with the few of these websites they checked.
2. In notepad.exe, lightly delete any noise rows at the beginning or end of the generated CSV.
3. Open the CSV file in Microsoft Excel.
4. Sort the redirection rows.
5. Convert the CSV rows back into JSON.
6. Store the newly sorted JSON entries back into the original single large `.openpublishing.redirection.json` file.

After the JSON file is sorted, it might be good to create and merge a PR at this interim point, before proceeding.

Indeed, you might want to convert your single redirection file to absolute paths, even if you do not presently want to subdivide it into multiple redirection files.

### 3. Cut and paste entries into new redirection files

For the first sorted group of redirection entries, cut and paste them into their own new dedicated redirection file. Remember to add exactly one dot-delimited node to the file name, immediately before the `.json` extension.

Ideally, store the new file in the same repo root directory where the original single file is still stored.

Repeat for other entry groups as appropriate.

Eventually, the original single redirection file might contain only a few groups too small to deserve their own separate redirection file, plus some miscellaneous entries.

### 4. Register the redirection files

In file `.openpublishing.publish.config.json`, add the `"redirection_files"` section. In this section, ensure every redirection file name is listed.

### 5. Create your PR

Now you can finish the whole process by creating a regular PR, and seeing it through to merge.

## Validation rules and messages

The following validation rules have been added to the build validation:

- *Error:* User registers a redirection file in `.openpublishing.publish.config.json`, but the registered file cannot be found in the specified location.
- *Error:* Two or more redirection entries are defined for the same source file, regardless of whether the two entries are in the same redirection file.
- *Warning:* A relative source path value starts with a forbidden `/` character, or an absolute source path lacks the required `/` as its first character.

## Localization implications

- All redirection files will be automatically synced to localized repos. Any updates to the existing redirection files will also be automatically synced.
- The configuration file (`.openpublishing.publish.config.json`) will be automatically synced to localized repos. However, updates to the configuration file, including updates to the redirection files registration section, have to be manually synced by loc PM team at this time. This part of the process will be improved with more automation by end of FY21Q3.

## Related links

- [How to redirect obsolete articles](#)
- [Validation message: redirect-url-invalid](#)

# Archived content

6/16/2021 • 3 minutes to read

[docs.microsoft.com/previous-versions/](https://docs.microsoft.com/previous-versions/)

All 45 million article across all MS business groups in MSDN and Technet library was reviewed and migrated in 2017-2018 to docs.microsoft.com or docs.microsoft.com/previous-versions. Content migrated to previous-versions content is not regularly maintained. Hence, was moved to what we consider our archive.

Each library was reviewed with the business owner and CELA to determine the correct location (docs.ms.com or previous-versions) and the banner to be used (archived or retired). Exclusive of age, all High-Volume Products (HVP) were migrated to docs.microsoft.com.

HVP list includes:

- Windows XP and forward (including .NET Framework)
- Windows Server 2008 and forward
- All versions of .NET Framework (not including .NET Core)
  - the previous three are under regulatory supervision
- Office 2007 and forward
- SharePoint 2007 and forward
- Exchange 2007 and forward
- SQL Server 2008 and forward

## NOTE

Visual Studio is not a HVP and topics may be removed without concern of legal antitrust compliance violations]

## Principles for Archived content

1. Archived content typically will be lower fidelity than what is on docs.microsoft.com.
2. Most of the content isn't indexed for search engines, except if it is an HVP product, that will be indexed.
3. Reference isn't auto-generated but migrated as .md
4. Our internal support team will respond to customer issues generally as a P2 unless it is a 404 or egregious site issue. Broken links or missing images are not a high priority issue under previous-versions. The actions to take would be to route the customer issues to the repo owner.
5. Content not in a table of contents (TOC), if relevant such as reference, was republished under previous-versions also not in a TOC.
6. Any reference content that has no actual content will be deleted. The legal council for HPV products has reviewed and recommended this action to reduce the risk of false negatives—a topic is available but incorrectly identifies an API as documented when it is not.
7. Any topic with pre-release disclaimer can be removed.
8. All migrated topics from MSDN/TechNet should be redirected to Docs target.

## Location of previous-version content

All content is stored in Azure DevOps repos typically by the name of the product.

- All repos are under

a) <https://docs-archive.visualstudio.com/docs-archive-project>

b) <https://docs-archive.visualstudio.com/office-dev-reference-archive>

## Permissions for managing previous-version content

All Microsoft FTEs have view-only permission to all previous-version repos and content.

- To edit and/or publish previous-version content, membership in one of the following AAD Security Groups(SG) is needed.
- Each SG has DevOps contributor-level permission to all previous-version repos and content. Join the appropriate SG through https://idweb based on your working group.

AAD SG	DEVOPS GROUP
CSS Archive Repo Contributors	Customer Support
CSS Content Editors - Motiv	Customer Support
CSS Content Motiv	Customer Support
dotnetcontent-sg	Dotnet Contributors
Dynamics SMB UA - FTE	Dynamics Contributors
BACX CE Team Extended	Dynamics Contributors
Marvel Docs Contributors	Office Contributors
DMC prototype Git Repo Access	Protocol Contributors
sccmdocs-previous-version	SCCM Contributors
Visual Studio Archive Docs Contributors	Visual Studio Contributors
Windows Previous Versions admins	WDG Contributors

- If you need contributor-level permission but don't see the AAD SG or DevOps group for your content team, contact one of the project administrators:

The screenshot shows the Azure DevOps interface for the [docs-archive-project]\docs-archive-project Team. The Members tab is active, displaying a list of four team members:

- Jennifer Mashkowski (jmash@microsoft.com)
- Jerry Song (jesong@microsoft.com)
- Ke Xu (kexu) (kexu@microsoft.com)
- Sudeep Kumar (skumar@microsoft.com)

## Process to update previous-version content

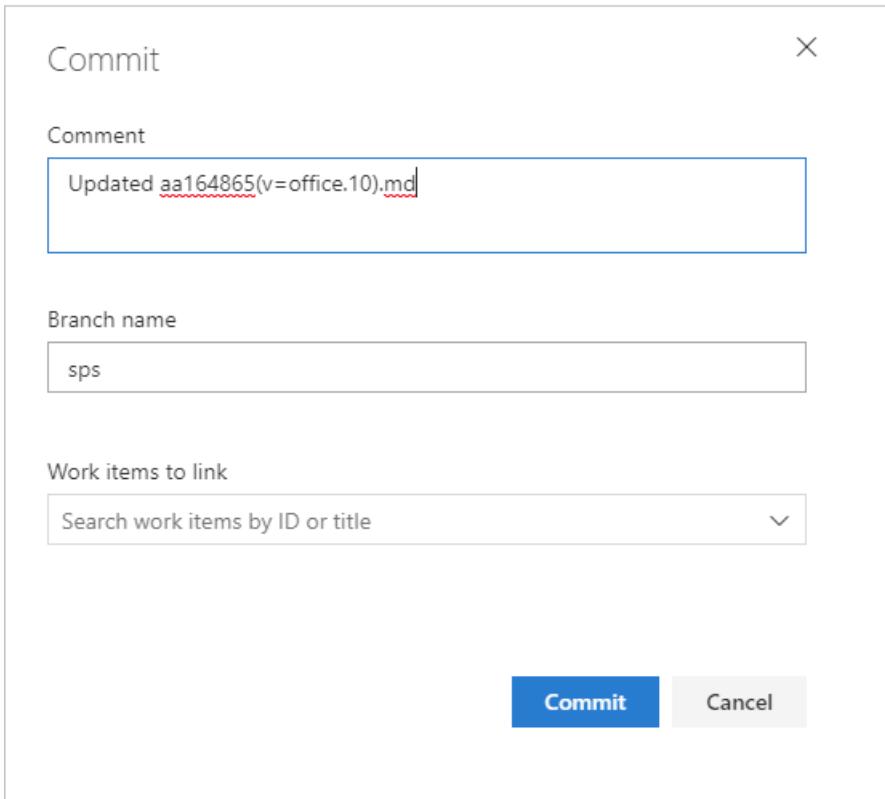
To make any critical updates to archived content, follow the below steps:

- Navigate to the Azure DevOps repo path of the file.
  - File location can be found at [Location of previous-version content](#)
  - File location can also be found from viewing source of the page from browser under meta name="original\_content\_git\_url"
- Content updates can be done directly from browser on Azure DevOps repo. After you navigate to the file,
  1. Select your working branch from the dropdown.
  2. Next step, edit file by selecting the 'Edit' pencil icon.

The screenshot shows the Azure DevOps repository interface for the 'office-dev-archive-pr' branch. The 'Edit' button in the top navigation bar is highlighted with a red box. The file 'aa164865(v=office.10).md' is open, showing its content and metadata.

title	TOCTitle	msassetid	msmtspurl	mscontentkeyID	ms_date	mtps_version
Microsoft Office 2000 Developer Star Page	Microsoft Office 2000 Developer Star Page	OfficeDevCoreJ-HTMLDestPage.htm	<a href="https://msdn.microsoft.com/en-us/library/aa164865(v=offce.10)">https://msdn.microsoft.com/en-us/library/aa164865(v=offce.10)</a>	3075346	06/4/2014	v=office.10

- After you complete the content edits on the md file, commit changes on your working branch.



- Create Pull Request against master branch and verify your changes.

docs-archive / docs-archive-project / Repos / Pull requests / office-dev-archive-pr

New Pull Request

From: sps into Into: master

Title \* Updated aa164865(v=office.10).md

Description Updated markdown to remove space

Add commit messages

Reviewers Search users and groups to add as reviewers

Work Items Search work items by ID or title

Create

Files (2) Commits (2)

- If changes look good, create PR from master to live branch and merge.
- You can verify build status on [Docs Portal page](#) from the [Sign in with Azure DevOps](#) option.

You can also submit CPS ticket [here](#) to update Docs previous version content.

# Art and multimedia gallery guidance

6/9/2021 • 2 minutes to read

## NOTE



THIS DOCUMENT IS IN REVIEW AND IS NOT YET SUPPORTED IN THE CONTENT STANDARDS FOR [DOCS.MICROSOFT.COM](#). WE ENCOURAGE YOU TO USE THE GUIDANCE AND PROVIDE FEEDBACK [IN OUR TWO-QUESTION SURVEY](#).

When you're looking for art or video to add to your Docs content, you may find several libraries of logos, icons, or videos. This article serves as a guide for understanding which libraries are updated regularly and recommended for use.

## Microsoft Brand Central

[Brand Central](#) is the hub for Microsoft's brand strategy. The site covers the principles of Microsoft's identity. You can learn about the brand's core elements too, like color, grid, illustration, photography, and more.

Use Brand Central if you need high-quality marketing materials, photos, logos, or guidance for a customer-facing scenario. Below are some of the assets available on Brand Central:

- [Microsoft and product logos](#)
- [High-quality photos](#) organized into collections (for example, Diversity and inclusion, Cloud, Microsoft campus life)
- [Microsoft-branded templates](#) for PowerPoint, Email, Word, Teams backgrounds, and more
- [Stories, elements, and resources](#) for each Microsoft product
- [Microsoft's branding strategy](#)

## Azure icons

Many sites host Azure icons and assets. However, not all of these sites are updated or maintained. Below is a list of asset libraries that you should use or avoid.

### Use

- [IconCloud](#) is the main icon repository for Azure assets. It contains thousands of icons, including those for MDL2, Bing, and Visual Studio. IconCloud is the most current and maintained asset library, with release notes posted at the top of the site each month.
- [Azure UX Patterns](#) provides supplemental Azure icons. The site also contains some Microsoft and product logos, though these might not be as up to date. If you're unsure whether a logo is the most current version, use Brand Central.
- [Azure architecture icons](#) hosts the official collection of Azure icons to help writers build custom architecture diagrams for their solutions. The bottom of the page provides a list of icon updates and the link to download the SVG icons.

### Don't use

- [Docs.microsoft.com shared image gallery](#): This library is no longer in use or maintained. Use one of the options above to find Azure-related icons.

## Video libraries

The Content & Learning video team makes custom content for its contributor customers. While the custom content follows style and other content guidelines, the team doesn't maintain a contributor-ready kit of video assets for all to use. They also don't maintain a central library or search area for specific video elements for contributor video.

If you need a video for your content, check out our guidance on [creating and publishing video](#).

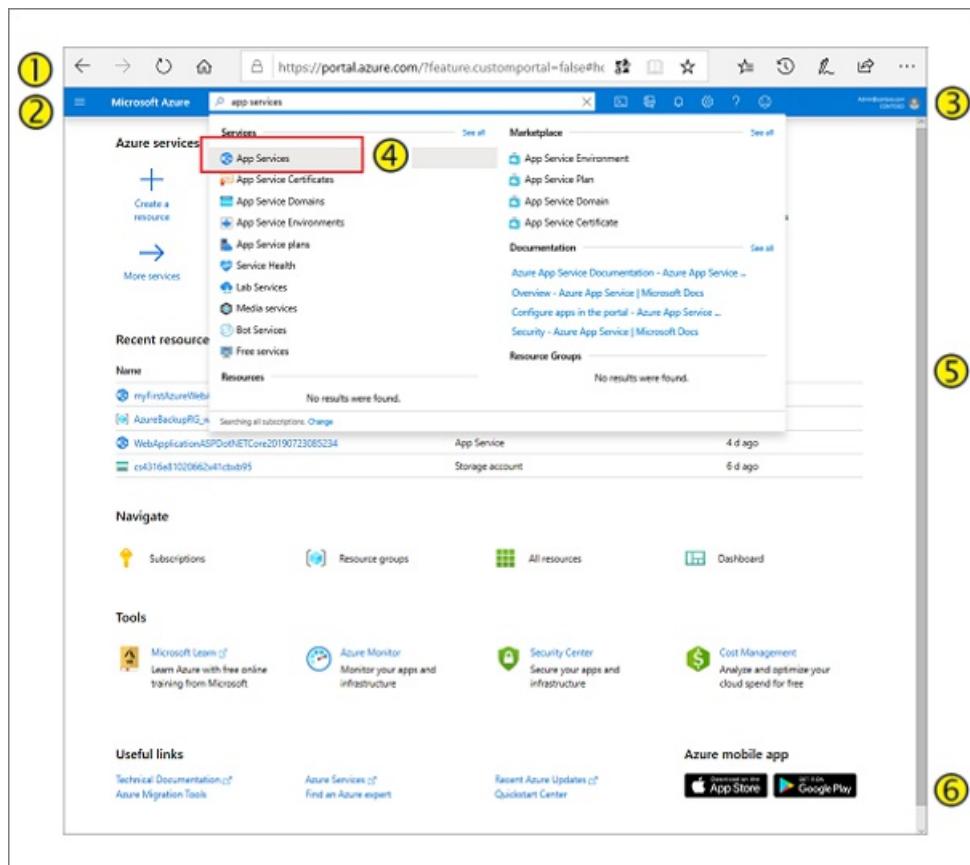
# Create and format screenshots in documentation

5/10/2021 • 14 minutes to read

This article provides information about screenshots: instructions for creating them, requirements for formatting them, and best practices for using them.

When screenshots are used in moderation, they clarify steps and provide helpful context.

## Elements of a screenshot



- ① For browser-based experiences, you must show the full window as your *first* screenshot of the article. The user sees that the experience is browser-based. Use the Safari (on Mac) or Chrome browser to highlight the cross-platform compatibility of Azure, where applicable. Include the browser frame, such as the URL bar and window controls.
- ② For IDE images, use default colors or [themes](#). For Azure portal images, use default colors and complete panes. The default theme in the portal is called *Azure*, with dark blue sidebars and a blue background.
- ③ Remove sensitive information.
- ④ Use a [red outline](#) to highlight a detail.
- ⑤ Use a [gray border](#) for screenshots with any light or dark edges to provide contrast for users of both light and dark themes. A gray border is added automatically if you use the Docs `:::image:::` syntax.
- ⑥ Resize the browser to minimize white space. Stretch the window to optimize or eliminate scroll bars when possible. Show full Azure panes when possible.

# Considerations for creating screenshots

Use screenshots sparingly, and only when necessary to clarify instructions. For example, use them for steps that aren't obvious. Using screenshots sparingly avoids content bloat and reduces the overhead of content updates.

Here are more points to consider as you create screenshots:

- **Mobile:** Avoid large image file sizes for low-bandwidth connections.
  - **Dimensions:** The Docs site automatically adjusts graphics to a certain maximum dimension based on the browser type (mobile or desktop). Image file dimensions shouldn't exceed 1200 pixels wide.
- The Docs site doesn't let you customize image size within Markdown pages, so edit the graphic file to the desired dimensions before uploading.

If you have images with large dimensions and readers need to see the detail, [create an expandable screenshot](#).

- **Localization:** Important articles can have screenshots localized for international consumption.
- **Alt text:** Convey the purpose. For example, if you want to show the features on an opening screen, you can use: "Screenshot of the main Lync window, showing the name of a person and icons for contacts, IM conversations, phone, and current meetings." Alt text should end with a period. For details, see the [alt text requirements for images](#) in the accessibility guidelines.
- **Border:** The gray border on light or dark image backgrounds is required for accessibility on light and dark themes. To learn how to make one, see the [section about gray borders](#) in this article.
- **Browser:** For content about core Windows technology, use the Microsoft Edge browser. Otherwise, use Safari or Chrome to show off cross-platform compatibilities.
- **Azure portal:** When you're creating Azure portal screenshots, hide pre-release features by using the `customportal=false` directive in the URL: <https://portal.azure.com/?feature.customportal=false>.

- **Browser frame:** When you're creating Azure portal screenshots, show the full browser frame in the first screenshot in each article to provide context to the reader. Subsequent screenshots in the same article can show a more focused view to avoid taking too much space.

## Tools for capturing images

Here are some common tools for taking screenshots:

- [Snipping Tool](#): Included on Windows 10.
  - Set **Mode** to do a rectangle snip, window snip, or full-screen snip.
  - To allow time to prepare menus and open drop-down lists before the snip is captured, set **Delay** to a few seconds.
  - To add a gray border to a snip, go to **Options**, set **Ink color** to **Gray**, and select the **Show selection ink after snips are captured** check box.
- [Apple macOS commands to take screenshots](#):
  - Press Command+Shift+3 for a full screenshot.
  - Press Command+Shift+4 for a screenshot of a selected region.
  - Press Command+Shift+4 and the Spacebar to take a screenshot of the current window.
  - Save screenshots as .png files on the desktop.
  - Use QuickTime to [record a screencast](#).
- [Ubuntu Linux](#):
  - Press the Print screen key for a full desktop screenshot.

- Press Shift+Print screen key for a screenshot of a selected region.
- Press Alt+Print screen key for a screenshot of a selected window.
- Press Ctrl+Alt+Shift+R to start and stop a screencast video recording.
- Screenshot files are saved in the Pictures folder, within your home directory. Videos are saved in the Videos folder.
- [Snagit](#): Licensed screenshot tool for Windows and Mac. To request a Snagit volume license key and provide business justification, contact your manager or admin for the Content & Learning team ([C + AI Business Operations Support](#)).
- [OneNote](#): Screen snipping tool.
- [Camtasia](#): Licensed screen capture and editor tool for Windows and Mac.
  - For Content & Learning team licensing, email [C + AI Business Operations Support](#).
  - For product team licensing, contact the team admin or business manager to obtain licenses. (Content & Learning does not fund those licenses.)
  - For more information, see the section about computers, peripherals, and software in the [Content & Learning BizOps team finance guide](#).

## Tools for editing images

To alter screenshots, you can use tools like these:

- [Paint]: Native app in Windows 10 that can do very basic edits to images.
- [Paint.NET](#): Free image-editing tool on Windows that uses the Microsoft .NET Framework.
- [GNU Image Manipulation Program \(GIMP\)](#): Free cross-platform image editor for GNU/Linux, OS X, and Windows.
- [Preview](#): macOS native tool that has basic editing capabilities like cropping, annotating, and drawing primitive shapes.
- [Adobe Creative Cloud Suite](#): Licensed editing suite for images and videos. It includes Adobe Photoshop. To request a license key and provide business justification, contact your manager or admin.
- [JS Paint](#): Web-based version of Microsoft Paint.

## Add an image in Markdown

There are two approaches to add images in Markdown: standard syntax and Docs syntax. See [Docs Markdown reference](#) for full details.

### Standard image syntax

The following code is the standard syntax for adding an image:

```
![Alt text that describes the content of the image.](/media/folder-with-same-name-as-article-file/service-technology-image-description.png)
```

### Docs image syntax

The Docs `:::image:::` syntax is custom Markdown for adding an image. We recommend this syntax over the standard Markdown for images because it provides more capabilities, such as the ability to add an image with a long description or a purely decorative icon.

The Docs syntax for a content image is as follows:

```
:::image type="content" source="/media/folder-with-same-name-as-article-file/service-technology-image-description.png" alt-text="Alt text that describes the content of the image.":::
```

## Converting from standard image syntax to Docs image syntax

To convert from the standard Markdown image syntax to the Docs image syntax, you can use regular expressions in the search panel in Visual Studio Code. In addition to the regex, search for `![ ]` to find other variations of markdown images that are too complex to replace automatically.

```
#Find images in markdown format like ![ ]( )  
!\\[(.+)\\]\\((.+)\\\)  
  
#Replace with new format  
:::image type="content" source="$2" alt-text="$1":::
```

## Create names for screenshot files

Here are some requirements and tips for naming screenshot files:

- Save screenshot files as .png (lowercase).
- Use a descriptive, lowercase file name. Avoid spaces (use dashes instead).
- Save the image files into the media folder adjacent to the Markdown file where they're rendered, in a subfolder that matches the article name. For example: `/media/article-name/description.png`.
- Avoid sharing screenshot files across multiple articles, because it can cause difficulties with broken image paths when files are moved or deleted.

For more information, see [File name and path guidelines](#).

## Resize images

Use 100% zoom when you're capturing screenshots on apps that are adjustable, and resize the images in an editor. Image dimensions should not exceed 1200 pixels.

To resize an image:

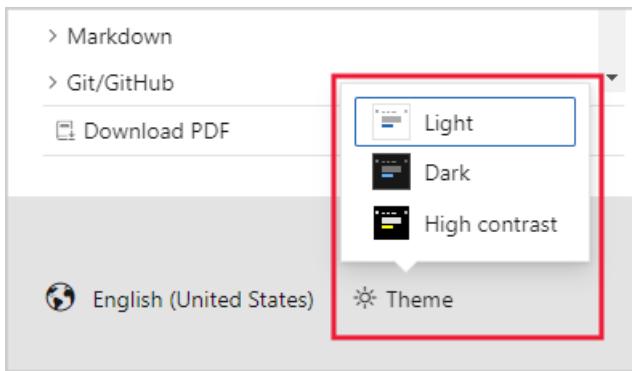
- In Paint.NET, select **Image > Resize**. Select **Maintain aspect ratio** to avoid distorting the image.
- In GIMP, select **Image > Scale Image**.

In Visual Studio Code, use the image compression feature of the Docs Authoring Pack to minimize the image file sizes. From the Explorer, right-click an image file or a folder and select **Compress image** or **Compress images in folder**.

## Test multiple themes

When you're working with conceptual art and icons, test with all three of our site themes: **light**, **dark**, and **high contrast**.

Images with backgrounds and transparency need to work with each theme. Conceptual art and foreground text can be unreadable on the Docs site if you use a dark theme with a transparent background. Transparency is useful in branded icons and other non-text artwork.



## Create a gray border

Use a 1-pixel gray border around all screenshots. Here's an example of a gray border around white space:

The form includes fields for Name, VM disk type (SSD selected), User name, Authentication type (SSH public key selected), and SSH public key.

If you use the Docs `:::image:::` syntax as shown in the following example, a border is applied via the `border` property. This property defaults to `true`, so you don't need to specify it explicitly. The image extension is the recommended way to add a border. Don't create borders manually in an image editor.

```
:::image type="content" source="media/image.png" alt-text="Alt text here.":::
```

To publish the image without a border, you can set `border` to `false`:

```
:::image type="content" source="media/image.png" alt-text="Alt text here." border="false):::
```

The border property is equivalent to the `mx-imgBorder` style for standard Markdown images:

```
> [&lt;div class="mx-imgBorder"&gt;  
> &lt;img alt="my-cool-graphic.png"/&gt;
```

## Create a red outline

To create a red outline, use 3-pixel width with red color #EF1B36 or RGB (239, 27, 54). Vertically center the element or text that you want to outline in red. Make the outline fill the shape.

Here's an example of the red outline used to highlight part of the UI:

The screenshot shows a configuration interface with several input fields:

- \* Name: A text input field.
- VM disk type: A dropdown menu showing "SSD".
- \* User name: A text input field highlighted with a red border.
- \* Authentication type: A dropdown menu with "SSH public key" selected and "Password" as an option.
- \* SSH public key: A large text input area.

To add a red outline by using GIMP:

1. Download [GIMP](#) and open the app.
2. Open the screenshot image. Select **File > Open**.
3. Set up the color palette to remember the custom red color. When you set up the palette the first time you use GIMP, the colors are saved for future use.
  - a. Double-click the black foreground color square in the toolbox window. The **Change Foreground** color picker opens.
  - b. In the **HTML notation** text box, type **EF1B36** and then press enter to see the red color. The current foreground color is now set to red.
  - c. Save the color into the color history by selecting the arrow icon. This color is saved for future use in the color palette.
  - d. Select **OK** to close the color editor.
4. Create a new layer by using the **Layers** pane. Select the **New** icon (paper sheet). Name the layer, and use transparency for the background. The new layer isolates the rectangles from the original background image in case you need to make adjustments.
5. Select **Rectangle Select Tool** from the toolbox (or press the R key). Draw the region in the image that you want to draw the box around.
6. From the **Edit** menu, select **Stroke Selection**.
7. Select **Stroke line > Solid Color**. Set **Line width** to 3.0 pixels, and then select **Stroke**.
8. To save the image, select **Export Image** and name the image by using .png format. Optionally, you can select **Save** to save the native .xcf GIMP format with the layers and other project metadata for future edits.

To add a red outline by using Paint.NET:

1. Download and open [Paint.NET](#).
2. Select **File > Open**. Select the screenshot file.
3. In the color palette, select **More > >**. Adjust the color by entering **EF1B36** in the **Hex** box. (You can also enter it manually as decimal R 239, G 27, B 54.)
4. Select the shape tool (or press the O key).
5. On the shape toolbar, select **Rectangle**, and then select **Draw outline** in the drop-down list for the shape fill mode.
6. Set **Brush width** to 3.
7. In the **Layers** palette, select the plus sign (+) to add a new layer. The new layer isolates the rectangle from the original background image in case you need to make changes.
8. Draw the red rectangle and adjust the dimensions and position as needed.
9. Save the file by selecting **File > Save**. Set **Save as type** to use the .png extension. Optionally, you can select

**Save** to save the project in the native Paint.NET format, .pdn. Layers and other project metadata will then be available for future edits.

## Capture the cursor

One easy way to capture the mouse cursor in your screenshots is to use the Snagit 2020 app.

To configure Snagit to capture the cursor:

1. Open the Snagit app.
2. Select the tab for the type of capture (**All-in-one**, **Image**, or **Video**), and then turn on **Capture Cursor**.

To use Snagit to capture the cursor:

1. Move your cursor to the spot.
2. Use the keyboard shortcut (default is Print screen) to capture the image with the cursor.

For more information, see the [TechSmith support article](#) about capturing an image.

## Remove sensitive information

Consider if your screenshot contains any of the following kinds of information:

- Subscription names: evaluate for private info & appropriateness
- Subscription IDs: must be replaced with fictitious info
- Usernames: should map to fictitious name guidance
- Email addresses: should map to fictitious name guidance for individuals and companies (re: the domain)
- Computer names: evaluate for private info & appropriateness

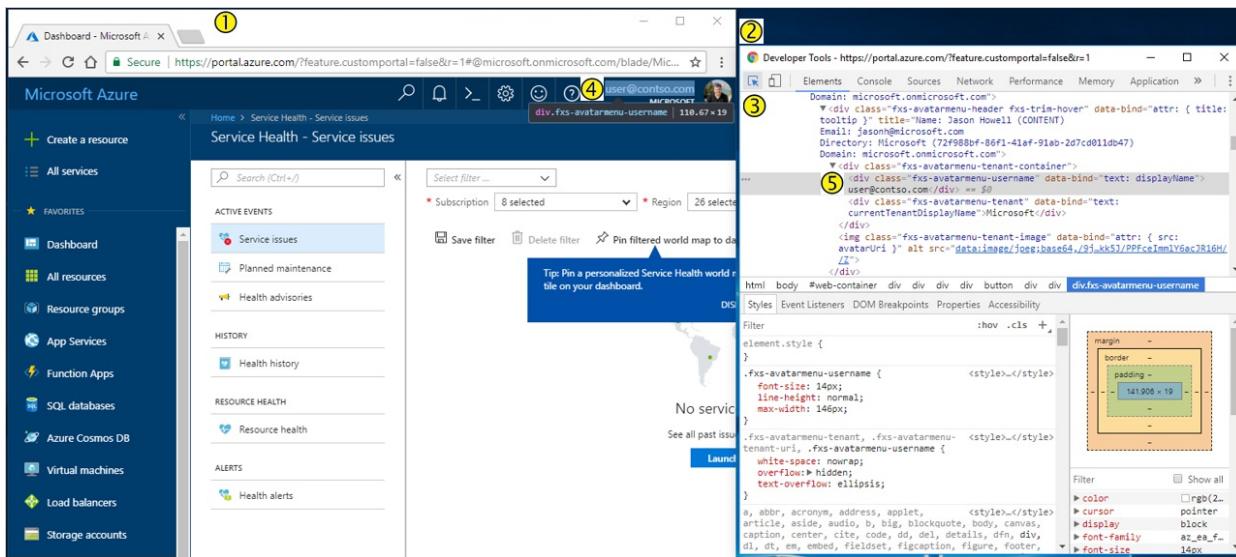
You should replace personally identifiable information with fictitious information, such as Contoso-based names and resources. Here are two resources to help explain acceptable values:

- Fictitious names, domains, and addresses from the [Microsoft Writing Style Guide](#)
- CELA's [approved fictitious names & guidelines](#).

In some instances, content contributors set up sanitized environments with fictitious values to collect screenshots. Otherwise, there are techniques to edit the screenshot to help minimize sensitive information, either by customizing the web page in the browser using developer tools or using an image editor to redraw portions of the screen.

### Remove information directly from a webpage

By using the browser developer tools in the Edge or Chrome browser, you can edit the HTML behind the text shown in the browser and remove personally identifiable information.



1. Open the browser and arrive at the page you want to capture.
2. Press F12 to open the developer tools in the browser. Undock the developer tools from the browser window to preserve the original screenshot.
3. On the toolbar for the developer tools, select the tool for selecting an element (pointer icon).
4. In the main browser window, hover over the information you want to change to isolate the element, and then click to select it. For example, select the username on the Azure portal.
5. In the developer tools, right-click the highlighted HTML tag. Select **Edit as HTML**.
6. Edit the HTML tags.

- Replace your account avatar with a generic one such as , which is available on the [Azure portal](#).

## Script webpage information replacement

You can script the replacement of personal information to save time from manually editing each browser HTML element.

In Chrome and Edge (Chromium) browsers, you can use script snippets that locate strings in the `<DIV>` and `<SPAN>` tags in the HTML and replace them with non-personal information. This works well for screenshots of the Azure portal with a minimal amount of code. Please share your feedback if this works in other portals. For more information, see [Run Snippets of JavaScript](#).

1. Browse to the portal page that you need to screenshot.
2. To open DevTools, press the F12 key. Alternatively, the ellipsis in the browser menu > **More tools > Developer tools**. Undock the Devtools tab into a separate window to give yourself more room.
3. On the **Sources** tab, select **Snippets**. If you cannot find it (it may be hidden to fit), expand the chevron `>>` icon on the left.
4. Create a `+ New snippet` and give it a friendly name.
5. Paste the following JavaScript code into the snippet editor. Edit the parameters in various `replace` function calls to represent your own name (as it appears on the Azure portal for example), company details, and account information. You can add additional elements to replace as long as the string is represented in the HTML page as a Div or Span tag.
6. To run the code, **right-click > Run** on the snippet name, or use keystroke **Control+Enter** in the code editor. The HTML page elements are changed and replaced with your fictitious values. Some errors may appear if the website code cannot parse the new values, but you can refresh the browser page (F5) if you need to start over or change pages.

```

// Chrome/Edge+ snippet to remove personally identifiable information.
// Tweak for name, email, subscription ID, etc.
// Install in snippets in Dev Tools

function replace(selector, text, newText) {
    var elements = document.querySelectorAll(selector);
    Array.prototype.filter.call(elements, function(element){
        if (text === element.textContent) {
            element.textContent = newText;
        }
    });
}

function replaceimage(selector, newImageUrl) {
    var elements = document.querySelectorAll(selector);
    Array.prototype.filter.call(elements, function(element){
        element.src = newImageUrl;
    });
}

// Provide your real values in the second parameter to be replaced with the fictitious values in the third
parameter
replace('div', 'Your Name', 'YourName')
replace('div', 'youralias@microsoft.com', 'user@contoso.com')
replace('div', 'abcdef01-2345-6789-0abc-def012345678', '{your subscription id}')
replace('div', 'Microsoft', 'Contoso, Ltd.')
replace('span', 'Your Name', 'YourName')
replace('span', 'youralias@microsoft.com', 'user@contoso.com')
replace('span', 'abcdef01-2345-6789-0abc-def012345678', '{your subscription id}')

// replace your real avatar image with a placeholder image
replaceimage('.fxs-avatarmenu-tenant-image',
'https://portal.azure.com/Content/static/MsPortalImpl/AvatarMenu/AvatarMenu_defaultAvatarSmall.png')

```

Example before:

1. Before

2. DevTools

Subscription \*: Azure Big Data and Beyond Docs Team

Resource group \*: JasonH

3. DevTools Sources tab open with a snippet named "Clean Screenshot".

4. New snippet and name it: Run code

5. Paste code and edit replace function parameters:

```

15 var elements = document.querySelectorAll(selector);
16 Array.prototype.filter.call(elements, function(element){
17   element.src = newImageURL;
18 });
19 }
20 replace('div', 'Jason Howell', 'YourName')
21 replace('div', 'jason@microsoft.com', '(your account)')
22 replace('div', 'e74f0ec1-631b-46f4-bde5-6aa999cf5be0', '(your subscription id)')
23 replace('div', 'Azure Big Data and Beyond Docs Team', 'Your Team')
24 replace('div', 'Microsoft', 'Contoso, Ltd.')
25 replace('div', 'JasonH', 'Contoso Resources')
26 replace('div', 'jasonHADF', 'MyFactory')
27 replace('span', 'Jason Howell', '(your name)')
28 replace('span', 'jason@microsoft.com', 'john@contoso.com')
29 replace('span', 'e74f0ec1-631b-46f4-bde5-6aa999cf5be0', '(your subscription id)')
30 replaceImage('.fxs-avatarmenu-tenant-image', 'https://portal.azure.com/Content/sta'
31

```

6. Run code

After script has run:

Your Subscription

Contoso Resources

## Use an image editor

If the information is not easily edited in HTML tags, use an image editor such as Paint.NET, Photoshop, or Snagit to cleanly draw over the personal information.

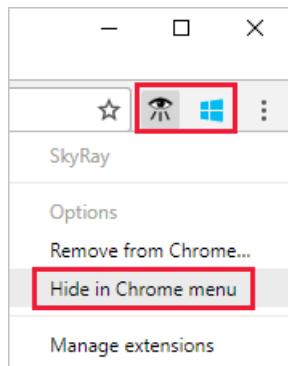
For example, use the color picker to select the background color, and draw a rectangle over the sensitive information. Then use a type tool to replace the information. Typically the portal uses Segoe UI font in dark gray or white, and size varies.

### Hide browser customizations

When you're using browsers, you might have customizations showing in your browser toolbars. Before you take a screenshot, hide those toolbars to remove personal information and avoid leaking internal tools.

For Chrome:

1. Hide your favorites toolbar by right-clicking the toolbar and clearing the check mark from **Show bookmarks bar**.
2. Hide any extension icons on the toolbar. Right-click each extension icon and select **Hide in Chrome menu**. The extension icon is then hidden, but it can be shown under the customization menu (vertical ellipsis, ...) as needed.



## Next steps

- [Add static art to your content](#)
- [Create an expandable screenshot \(lightbox\)](#)
- [Create and publish animated GIFs in documentation](#)

# Create an expandable screenshot (lightbox)

11/2/2020 • 3 minutes to read

In this article you will:

- Learn what a lightbox is and how it can be used.
- Receive guidance for how to appropriately leverage lightboxes.
- Learn how to create a lightboxed image.

Screenshots are commonly used in documentation. However, a screenshot of an entire webpage tends to be large. Putting the entire image inline in documentation can disrupt the flow of the page and impair readability.

Lightboxes to the rescue! A lightbox is a special way to display an image on your page. When users click a lightboxed image, an expanded image overlaid over the rest of the page content will appear. This allows you to use scaled-down or cropped versions of images inline, while still allowing users to access the larger version of the image if they desire. Users who don't need to see the entire image don't have their experience disrupted and users who do can view it without leaving the page.

There are two ways to use this feature:

1. Use two different images for the inline and expanded images. When users click on the inline image, a lightbox with a different image will appear. This is appropriate for images that can be cropped to highlight the most important part of a webpage or application inline and then displayed in full in the lightbox.
2. Use the same image for the inline and expanded images. When users click on the inline image, a lightbox with a larger, zoomable version of the same image will appear. This is appropriate for images that will lose important context if cropped and need to be shown in full inline.

## Examples

To see this feature in action, we'll be looking at a page from Xamarin's documentation: [Setup and Installation](#). Click any of the following images to see the lightbox in action.

### NOTE

For the sake of clarity, the image filenames and paths as used on this page have been changed from the filenames and paths used in the original Xamarin article.

### Using two different images

Here's an example of using a cropped image inline, and the full image expanded. The cropped image highlights the most relevant part of the full image.

## Choose a template for your new project

The screenshot shows a mobile application template selection screen. On the left, there's a sidebar with categories: Multiplatform (App, Library, Tests), iOS (App selected, Library, Tests), and Android (App, Library, Tests). The main area is titled "General" and contains five items: Single View App (selected, indicated by a blue background), Master-Detail App, Tabbed App, Page-Based App, and WebView App. Each item has a small icon to its left.

Using a lightbox here allows for significantly improved page flow. Users who want to better orient themselves by viewing the full interface can do so with a single click, and users who don't need that context don't have their experience disrupted with an overly large and verbose image.

### Using the same image

Here's an example of using a larger version of the inline image as the expanded image:

This screenshot shows the same template selection interface as above, but with a larger, detailed image of a WatchKit App. The image features a smartwatch with a circular face and a rectangular interface overlay, likely representing a watchOS extension. The main interface elements are identical to the first screenshot, including the sidebar and the "General" section with its five template options.

In this example, the whole interface is relevant to the task the user is trying to accomplish, so cropping the image would be unhelpful. Using a lightbox here allows the user to zoom in and get a better view of the text, which may be difficult to read inline as the image is smaller and not easily zoomable.

## Feature Usage

Lightboxes on Docs are composed of an image linked to another image.

### How to add a lightbox with the Docs image extension

1. Add your inline image:

```
:::image type="content" source="image-file-inline.png" alt-text="Image alt text":::
```

2. Add the `lightbox` attribute. The value of `lightbox` is the path to the expanded image.

```
:::image type="content" source="image-file-inline.png" alt-text="Image alt text" lightbox="image-file-expanded.png":::
```

Using `:::image:::` saves several steps compared to standard Markdown.

## How to add a lightbox with standard Markdown

1. Add your inline image:

```
![Image alt text](image-file-inline.png)
```

2. Wrap that image in a link by adding brackets `[ ]` around it and parentheses `( )` after:

```
[ ![Image alt text](image-file-inline.png) ]()
```

3. Set the link location inside the parentheses `( )` to the path where your expanded image file lives. If you're using one image for both your inline and expanded images, this will be the same image path you linked to in step #1:

```
[ ![Image alt text](image-file-inline.png) ](image-file-expanded.png)
```

or

```
[ ![Image alt text](image-file-inline-and-expanded.png) ](image-file-inline-and-expanded.png)
```

4. Append `#lightbox` to the path where your expanded image file lives. This tells Docs that when a user clicks the inline image, a lightbox with the expanded image should appear over the page:

```
[ ![Image alt text](image-file-inline.png) ](image-file-expanded.png#lightbox)
```

### IMPORTANT

Adding `#lightbox` to the end of your expanded image path is the most critical step. If you don't include this, when a user clicks the inline image they'll be redirected away from the current page and to a new page with the expanded image, just like if they clicked a normal link.

Let's put it all together with one more lightbox example, this time with a cute kitten:



```
[ ![Here's a cute kitten](media/contribute-how-to-use-lightboxes/cute-kitten-inline.jpg)](media/contribute-how-to-use-lightboxes/cute-kitten-expanded.jpeg#lightbox)
```

## Feature Guidance

For general guidance on using images, see [How to add static art to your content - naming and placement of media files](#).

Inline images should be suffixed with `-sml` or `-inline`. Expanded images should be suffixed with `-lrg` or `-expanded`.

Descriptive alt text should always be provided for users with screen readers.

# Conceptual art

5/10/2021 • 4 minutes to read

Use conceptual art to simplify complex subjects for the reader. It's especially helpful in introductory articles. Keep in mind that art occupies screen real estate, so use it judiciously.

## Standards

These standards ensure a consistent look and meet other requirements:

- Two-dimensional
- No borders
- Correct size
- Current Microsoft style for icons, fonts, colors
- Legal and accessible

## Get help from the Content & Learning art vendor

While most content contributors make their own screenshots and basic artwork, if you have a complex art requirement, you may need assistance. Content & Learning contracts with a graphics artist who can assist with screenshots and other conceptual art. Seek assistance through the ticketing process as follows:

### To seek assistance from the artist:

1. Create a work item in the Azure DevOps board corresponding to your project to track the request:
  - Docs related projects: [template](#)
  - Learn related project: From your ModuleWorkItem, select the green "New Conceptual Image" button in the OneClick actions to create a new image workitem that is associated with the corresponding Learn module. Assign that item to v-deste@microsoft.com. If needed, here is a [template](#).
2. Attach any type of draft (architecture, image, etc.) in the work item.
3. Then email the artist with the item number: Deborah Steinke at v-deste@microsoft.com

## Do it yourself: Follow these standards and practices:

- **File types:** For conceptual art (screenshots are best as PNG's)
  - **SVG:** SVG is the preferred file type for conceptual art because it scales without loss of clarity regardless of the browser window. SVGs do not, however, currently render in PDFs. If the art includes text then ask your team artist to convert the text to SVG (converting the text is an extra step). **Important:** Do not use interactive or scripted SVGs on Docs (for design, quality, and security reasons)
  - **To repo admins:** Make sure the SVG file type is in your docfx.json file, or the image will not appear in the published file. At the end of the "files" section add: "\*\*/\*.svg"
- **PNG:** Acceptable format. As a raster format (pixels), if the art is viewed at its original saved size (100%), then it is sharp. But sharpness is lost when the browser window reduces the art to other than its 100% size

- **File names:**
  - Use a lowercase file name. Use a descriptive name. Avoid spaces (use dashes or underscores instead)
  - Save the image files into the accompanying media folder, or in a subfolder matching the article name  
For example, /media/article-name/description.png
  - Avoid sharing art files across multiple articles, since it can cause difficulties with broken image paths when files are moved around or removed completely
- **Image type:** Docs Markdown supports standard conceptual images, complex images that require longer descriptions, and decorative icons. Each has different requirements and capabilities. Choose the right syntax for your image. For more information, see [Images](#) in the Docs Markdown Reference.
- **Alt text:** Alternate text describes an image so that readers with visual impairments, including those using screen readers, can understand the purpose of the image. Alt text is an accessibility requirement for conceptual and complex images. Complex images also require a long description to fully communicate the intent of the image.

Because icons are purely decorative, alt text is redundant and should not be used. If you get a build validation warning for missing alt text on an icon, the solution is to use the image syntax for icons as described in [Images](#).

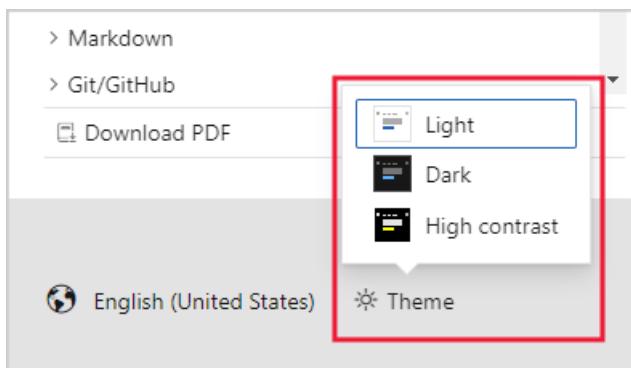
For more information about writing good alt text, see [Accessibility Guidelines for Multimedia](#).

- **Localization:** Alt text is also helpful for localization, as is summarizing a complex image in a long description, or any image in surrounding text. If your image belongs to a different product than the documentation that contains it, you can ensure accurate localization by specifying the `loc-scope` attribute on the image.

### Test multiple themes and transparency

When working with conceptual art and icons, test with all 3 of our docs.microsoft.com themes: **light**, **dark**, and **high contrast**.

Images with backgrounds and transparency need to work with each theme. Conceptual art and foreground text can be unreadable on the Docs site set to the dark theme if a transparent background is used. Transparency is useful in branded icons and other non-text artwork.



- Correct use of transparency on icons:

**APPLIES TO:**  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

- Incorrect background on icons:

**APPLIES TO:**  SQL Server  Azure SQL Database  Azure SQL Data Warehouse  Parallel Data Warehouse

## What about the source files for conceptual art?

To aid in future maintenance of the conceptual art as the associated document evolves, make it possible for authors to obtain the source file, such as Visio or Adobe Illustrator files. If possible, commit the source file to the same repository in the same location as the SVG or PNG image. If that is not possible, and the source file resides in some other official location, such as a public GitHub repo managed by Microsoft or a partner, include an HTML comment with the fully qualified link to the source file.

For example, an article uses an SVG file for conceptual art, but the SVG file was generated from an Adobe Illustrator file.

```
<!-- Source for the following diagram is at https://github.com/Azure/azure-sdk-for-java/blob/master/sdk/spring/azure-spring-doc-resource/spring-to-azure-keyvault-certificates.ai -->
:::image type="content" source="media/configure-spring-boot-starter-java-app-with-azure-key-vault-certificates/spring-to-azure-keyvault-certificates.svg" alt-text="Diagram showing interaction of elements in this tutorial." border="false":::
```

## Next steps

- [Screenshots: How to create, format, and embed in content](#)
- [Microsoft Azure, Cloud and Enterprise Symbol / Icon Set - Visio stencil, PowerPoint, PNG, SVG](#): This set of graphics is not complete but is still useful.

# Create and publish animated GIFs in documentation

10/31/2019 • 6 minutes to read

This article describes considerations for using animated GIF graphics, and helpful tools and approaches to make animations. Here are considerations and guidelines for creating animated GIFs:

## Why use animated GIFs?

- They can instruct users, illustrate problems for customer support, or mock up a new idea for the design process.
- They are easy to create.
- They are natively supported by effectively all browsers, unlike many video formats and players, so your audience doesn't need to install a plug-in to see it.
- Compared to a sequence of still images, they are less cognitively demanding. Your viewer doesn't have to mentally interpolate frames between still images because the frames actually exist. For example, trying to follow step-by-step drawings of how to tie a knot can be much more challenging than watching an animated equivalent.
- They can be embedded in most communication and be reasonably expected to work. So you can use a GIF in a technical document as easily as in a tweet or an email.

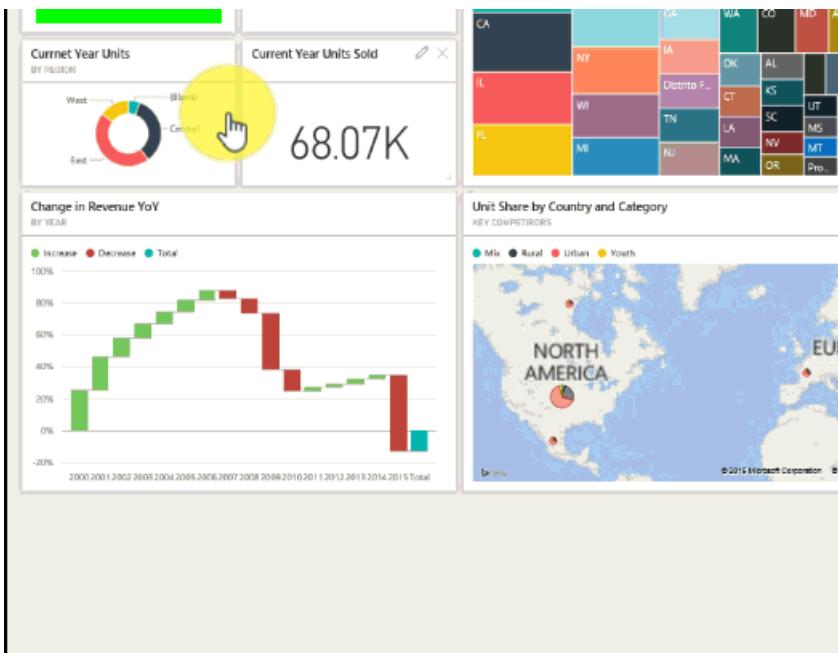
## Guidelines for using animated GIFs

Animated GIFs should be:

- **Used sparingly:** Space animated GIFs so only one displays on screen at a time. Too much visual noise hinders a user's ability to appreciate anything on screen.
- **Small:** GIF isn't an efficient image format. It is a bitmap without any image compression. Therefore large animations can cause the page load time to slow down. As you create them, notice your .gif file sizes, and optimize as needed. The color palette can be compressed (256 colors, etc), and the animated files can be optimized to help minimize the file size.
- **Short:** If a GIF goes longer than 5 seconds, video may be a better format to pursue.
- **Demonstrative in nature:** The GIF should show what something is or what it does, but without the expectation that the viewer is meant to be able to reproduce the demonstration themselves.
- **Used for familiarization:** It should acquaint the viewer with something that they're going to do themselves, but with the expectation that more detail is forthcoming.
- **Alt text:** Animated GIFs should contain alt text, just like you would enter for an image. Alt text should provide the text-equivalent to any non-text visuals included in the GIF, or anything that a visually impaired user would need to understand the content without being able to see it. Including alt text will also help animated GIFs to show up in search results and will help localization. See [Alt text](#) for guidelines.
- **Localized:** Consider that it is difficult to localize animated GIFs into multiple languages. Most likely the animated GIF is not localized at all. If your content is loc sensitive, consider using a static image so that it can be localized.

## Examples of animated GIFs in documentation:

- Power BI documentation uses animated GIFs for demonstrations in the documentation. For example:



- [Power BI example in context](#)
- [Power BI example 2](#)
- Visual Studio Code uses a number of animated GIFs for demonstrating user interface gestures: [Visual Studio animated GIF example](#)

## Free tools with steps

The easiest way to make a gif is to use the game bar built into Win10 to create an .MP4 file, and then convert it into a gif using the open-source converter [FFmpeg](#).

### Enable the game bar

First, you'll want to make sure that the game bar is enabled from the Windows Settings.

1. Search for `game bar` in the start menu and select **Game bar settings**.
2. Select **On** under **Record game clips, screenshots, and broadcast using Game bar**.

### Record your screen

Next, you'll want to access the game bar and record your screen.

1. Open the application you want to record (such as Microsoft Edge).
2. Use **Win+G** to open the Game bar overlay.
3. Select the recording icon to record the selected app.
4. Perform the action you want to record and then select the **Stop** icon on the recording overlay.
5. Select the **Game clip recorded** message that appears on your screen to access the recording .MP4. The default location is `%userprofile%\Videos\Captures`.

### Edit the .MP4

You can use the built-in video editor to trim the .MP4. Just search for `Video Editor` in the start menu, and then drag and drop your .MP4 file into the pane on the right to edit your video.

### Use FFmpeg to convert

Once your .MP4 is ready, you can convert it to a .GIF using the open source program FFmpeg.

1. Download [FFmpeg](#) and extract it.
2. Copy your .MP4 into the same folder as the FFmpeg.exe.
3. Open an administrative command prompt and navigate to this location.

4. Run the following command to do the conversion, modifying the file names as necessary:

```
ffmpeg -i file.mp4 -b:v 1024m -bufsize 1024m file.gif
```

## Paid Tools

The following tools are useful for creating animated GIFs:

- **Camtasia:** Record a video in TechSmith Camtasia. Review this [intro youtube video](#) and the [Step by Step tutorial](#). Contact your manager and admin to get a Camtasia license key.
- **Snagit:** Record a video of your screen using Snagit. Then use the editor to extract the animated GIF from the video. Follow their [online documentation](#). Contact your manager and admin to get a Snagit license key. CSI department contact: Indira Smith
- **Photoshop:** For an excellent outline of animated-GIF-creation methods, tools and optimization techniques, including how to create a GIF with Photoshop, see [CSSTricks "Makin GIFs"](#). Contact your manager and admin to acquire Photoshop in the [Adobe Creative Cloud Suite](#).

## Online services:

There are a number of online services that help edit and optimize GIF files. Use these sites with discretion since they are not Microsoft sites, and do not guarantee confidentiality:

- **Screen capture:** [LICEcap](#) captures part of your screen and saves the result as an animated GIF.
- **Converting still images to GIF:** [ImageMagick](#) is a suite of tools for creating and manipulating images. The animate command can be used to stitch together many individual still images (for example, a series of screenshots or photos) into an animated GIF.
- **Optimizing file size and performance:** [Online GIF Optimizer](#) can help you reduce your GIF file size and allows you to choose the level to optimize color, frame rate, and lossyness. [Gifsicle](#) is a command-line tool that can significantly reduce your GIF's file size (also see the [documentation manual](#)). Lastly, this tutspplus article outlines [10 Ways to Optimize an Animated GIF File](#).
- **Setting the number of times a GIF repeats:** [Gifsicle](#) has a feature baked in to enable you to set the loop count (see these [instructions](#)).

## Adding code to enable start/stop capability for GIFs

Currently docs.microsoft.com does not support stopping animated GIFs, but that feature is encouraged for accessibility purposes. If you own the source code to the web site you are publishing the GIF files on, you could potentially add the following kinds of enhancements to improve the user experience with animated GIFs:

- **Play animated GIFs on click:** [Tutorial](#), with an [example page](#) showing this method in use, and the [code in GitHub repo](#) (requires jQuery dependency)
- **Play animated GIFs on hover:** [Tutorial](#). Requires [Embedly API](#) – lots of ways to include.
- **Add via plain vanilla JS:** Haven't found a code example yet (this [stackoverflow article](#) explains how to load a static image, and then switch to animated GIF on hover).
- **Pause/play GIF on hover with jQuery** ([Codepen](#))

## Accessibility considerations

W3C Web Content Accessibility Guidelines outline two sections that should be considered when using animated GIFs:

- [Guideline 2.2 - Enough Time](#): Includes guidelines around "Pause, Stop, Hide" and "Auto-updating."
- [Guideline 2.3 - Seizures](#): Do not design content in a way that is known to cause seizures.

Animation can cause significant distractions for users with cognitive disabilities. For people with a high degree of motion sensitivity, flash rates between 2 hz – 55 hz can cause an epileptic seizure.

In regard to blind or visually impaired users, "Alt" tags or content text should supplement any animated GIF explaining any content being demonstrated in the animation.

To address these issues, developers must provide accessible, non-animated methods for disabling or pausing animated content within a page.

## Related articles

- [Screenshots: How to create, format, and embed in content-](#)
- [Use as a new way of good app presentation \(Slides Framework\)](#)
- [GIFs as Documentation](#): A short guide to using GIFs in documentation and as documentation.
- [Designing Safer Web Animation For Motion Sensitivity](#): Outlines reasoning for accessibility concerns, how GIFs can affect people with motion sensitivity, and guidelines for designing safer motion on the web.

# Create and Publish Video

5/4/2021 • 6 minutes to read

The [C&L Video + Media Team](#) is here to help you successfully create, publish, and update video, motion design, and digital media for Microsoft Learn and docs.Microsoft.com. Ready to get started? Submit your Video, Motion Design, Animation, or other visual media request here: [C&L Video + Media Requests](#)

## Publishing videos

### RedTiger

RedTiger is the video hosting backend for Microsoft's OnePlayer, and the preferred video host for embedding video on Learn and Docs. RedTiger provides complete support for accessibility and captions in multiple languages.

The RedTiger team provides [documentation and tutorials](#).

The Video + Media Team will provide publishing support to RedTiger for videos created by, or in partnership with them. For all other videos, see [RedTiger support](#) for complete support details and instructions.

### Channel 9

Channel 9's publishing system provides another video hosting option that also supports our accessibility and captioning requirements. However, at this time Channel 9 publishing has been limited to Docs. Upon request, the Video + Media Team will provide publishing support to Channel 9 for Docs videos created by, or in partnership with them.

To open a request with the Video + Media Team to publish a video on RedTiger or Channel 9, visit [C&L Video + Media Requests](#).

### Other hosting platforms

It may be appropriate to publish a video in multiple locations. For example, you may also want to publish your video on Twitter, Instagram, YouTube, or other Social Media, linking back to Learn and/or Docs. The Video + Media Team will assist here, as requested.

#### NOTE

Microsoft Learn does not support the embedding of YouTube video. This is primarily for content availability reasons (YouTube is not available in China), as well as inability to control suggested videos, cookies, and URL changes. We also strongly discourage the use of YouTube on Docs, due to these and additional Microsoft Accessibility concerns.

### Embed a video in markdown

You can embed a video in your topic by using the following markdown syntax:

```
> [!VIDEO <embedded_video_link>]
```

#### RedTiger example:

```
> [!VIDEO https://www.microsoft.com/videoplayer/embed/RE1XVQS]
```

#### Channel9 example

```
> [!VIDEO https://channel9.msdn.com/Shows/Docs-Learn/Data-Serialization-Languages/player?format=ny]
```

These will be rendered as:

```
RedTiger example
```

```
<iframe src="https://www.microsoft.com/en-us/videoplayer/embed/RE1XVQS" width="640" height="320"  
allowFullScreen="true" frameBorder="0"></iframe>
```

```
Channel9 example
```

```
<iframe src="https://channel9.msdn.com/Series/Youve-Got-Key-Values-A-Redis-Jump-Start/03/player" width="640"  
height="320" allowFullScreen="true" frameBorder="0"></iframe>
```

#### IMPORTANT

The CH9 video URL should start with https and end with /player.

## Process to create a video

At a high level, the process of creating a video for Learn or Docs includes:

- The idea of and goals for a video. The Video + Media Team will assist, as needed, during the ideation phase.
- Fill out a short online [C&L Video + Media Requests](#) form.
- Write a script (if needed) with assistance from the Video + Media Team. To get started, here is a [video script template](#) and some [sample scripts](#).
- Record your screen capture (if needed) with assistance from the Video + Media Team. They will provide you with a free Camtasia license to do so.

When the script is final and approved, provide it to the Video + Media Team, who will:

- Create the video
  - This may include: storyboards, graphic design, filming, editing, animation, motion design
- Record and add professional voiceover to video
- Incorporate your Film-at-Home footage, as needed
- Lead review rounds with designated partners and stakeholders
- Caption and publish the final video to the appropriate channel

From there, you will be able to embed the video into Learn or Docs. Please note that the video requestor is responsible for content currency and all future video update requests.

## Uploading new videos

Any new videos should be uploaded using the following process:

1. Go to <https://aka.ms/VideoUploadRequest> and fill in the details for your video. You will need (note that none of these items will be visible to the public):
  - a. A title for your video.
  - b. A list of products/services that your video is related to.
  - c. The target page or (if you don't have the page yet) docset that your video will be hosted on.
  - d. A link to the MP4 file for your video (if you don't have a location to put the file, you can put it here temporarily: `\scratch2\scratch\apex`). MP4 files should be 720p or higher.

- e. A description of the video.
2. Submit (save) that item.
3. Within two business days, the video will get uploaded. The link you need for embedding will be placed into the work item, and it will be resolved *back to you*.
4. Once you have grabbed the video link, close the work item.
5. The video link can then be added to your post, using this syntax:

```
> [!VIDEO https://www.microsoft.com/en-us/videoplayer/embed/RE1XVQS]
```

## Video requirements

The Video + Media Team will ensure that your video meets or exceeds all Microsoft, brand, production, and accessibility requirements. In general, these requirements include:

### Video

- 1920 x 1080 HD video format
- Frame rate of 29.97 frames per second
- Microsoft-approved intros and outros
- Microsoft-approved branding elements and logos

### Visuals

- Screen captures must be done with hardware capture, Camtasia, OBS, or similar professional software, at 16:9 resolution. The Video + Media Team will provide a free Camtasia license, upon request.
- All text on a screen capture must be visible on different screen sizes, and have a high-contrast color scheme.
- All visuals must meet CELA legal requirements for content and branding.
- Animations must be professionally created using professional design and 3D tools.
- Live action footage must be professionally lit, framed, and filmed using professional equipment.

Film-at-Home video segments must also follow Microsoft and Video + Media Team guidelines.

### Audio

- Professional audio that is free of echo, distortion, noise, and clipping.
- Voiceovers done by professional voice actors, featuring a diverse and global set of voices.
- American Disabilities Act video captions are required. The Video + Media team can assist you, upon request.
- An audio description isn't required at this time, but is highly recommended. Contact the Video + Media Team for more information.

## Video styles

In general, the Video + Media Team offers four main styles of videos (a video may sometimes mix these styles, as needed).

VIDEO TYPE	DESCRIPTION	ADDITIONAL DETAILS	SAMPLE
Screen capture	Visuals are screen captures from your computer	<a href="#">Making a screen capture video</a>	<a href="#">How to use the Azure Cognitive Services .NET Speech SDK for recognition</a>

VIDEO TYPE	DESCRIPTION	ADDITIONAL DETAILS	SAMPLE
Animation	Animation and motion design are used to explain difficult concepts, such as data flowing to the cloud	<a href="#">Making an Animated Video</a>	<a href="#">What is cloud computing</a>
Live action	One or more people speaking to a camera, interview-style	<a href="#">Making a One Dev Question video</a>	<a href="#">How did Xamarin get its name and why do you all love monkeys so much?</a>
Film-at-home	Record yourself speaking to a camera, interview-style, at home	<a href="#">Filming at home guide-doc</a> <a href="#">Filming at home guide-video</a>	Coming soon

## Video types

Within these video styles, there are several different video types that you can create. Select the video type that matches the context for how the video will be viewed.

For example, if the video will be used at the beginning of a Learn Module, the viewer will expect more of an overview of the Module concepts. If the video is embedded deep within a Doc, the viewer will expect the video to be more detail driven and direct.

VIDEO TYPE	DESCRIPTION
Conceptual	Conceptual overview of a technology or feature
How-To	A walkthrough of how to use a technology or feature
Sample Showcase	Overview of a sample and what a developer could learn from it
Design	Help designers understand how to implement UI features
Promotional	Help promote a developer technology, product, or platform

## Submit your Video + Media request

The C&L Video + Media Team is here to create and publish your most impactful and engaging video yet. To submit a request, head to the [C&L Video + Media Team Online Request Form](#).

## Next steps

- [Interpret data and customer feedback](#)

# Diagrams for technical communicators

4/7/2021 • 2 minutes to read

In this how-to guide you will draw technical diagrams, provide feedback to existing diagrams, and prepare your diagram as an illustration. This set of articles provides insight and recipes for the visual documentation of technical subjects. You can follow these steps even if you're uncomfortable with your ability as an illustrator. If you're an artist, you can find help creating technical illustrations. With this guide and help from Microsoft's graphic professionals, you can produce complete, correct, and appealing diagrams and graphics.

## Draw technical diagrams

Use the following articles to find guidance for creating and using formal diagram formats.

ARTICLE	DESCRIPTION
<a href="#">How to choose a diagram type</a>	You can use this decision tree to find the diagram type, diagram instructions, and templates to create a diagram.
<a href="#">How to draw a diagram from scratch</a>	Follow these steps to draw a diagram on a blank white board, sheet of paper, or application pasteboard.
<a href="#">How to draw a diagram using a diagram type</a>	Follow these steps to choose a diagram format to capture a technical concept.
<a href="#">How to provide feedback to a diagram</a>	Writers often receive a diagram that has already been drawn. These steps will get you started in reviewing the diagram, providing feedback, and incorporating the document in your diagramming workflow.
<a href="#">Diagram cookbook</a>	This section is a collection of diagram types. Each diagram includes the ingredients (things you need to get started) and instructions on how to draw the diagram.
<a href="#">Process mapping</a>	A process map is a picture of steps, decisions, and other elements of either a work or mechanical operation. This section provides tools for defining and summarizing a process.
<a href="#">Diagrams as code</a>	An alternative diagramming workflow takes the separation of diagram from presentation a step further. With diagrams as code, you can describe your diagram in a text format such as GraphViz's DOT language or Mermaid's markdown syntax.
<a href="#">About technical diagrams and art</a>	Learn about different categories of visualizations that you can use in your documentation. There are different approaches to how you might think about art in technical content.

## Next steps

Learn more about using art in your content and read [Conceptual art](#)

# How to choose a diagram type

4/7/2021 • 3 minutes to read

This article provides a set of questions to help you select the diagram type for your subject.

- Is your subject [software or hardware](#)?
- Is your subject [conceptual or made of data](#)?
- My subject is [something else](#).

## General visualizations

- Are you describing [the relationship of concepts](#) to each other?
- Are you [summarizing and visualizing data](#)?
- Are you describing the [interaction of actors doing work](#)?

## Concept diagrams

- Are the primary elements in your subject things and their relationship to each other?  
Then, [use a logical class diagram](#).
- Are the membership of subelements to two or more categories critical to understanding the subject?  
Then, [use a Venn diagram](#).
- Is the proximity of elements key to understanding the subject? For instance, do you need to draw a map?  
Then, [use a network diagram](#).
- Is the relationship between hierarchies of elements key to understanding the subject?  
Then, [use a tree diagram](#).

## Data visualizations

- You can find guidance on choosing a data visualization chart type at [Data Visualization: Choosing a Chart Type](#)

## Process mapping

- Are you drawing a business process?  
Then, [use Business Process Modeling notation](#).
- Are you drawing the action of agents, their decisions, and the transformation of input into output?  
Then, [use an activity diagram](#).
- Are you drawing the lifecycle of something in the context of a system?  
Then, [use a state machine diagram](#).

## Software and systems

- Are you the [mapping interaction](#) of actors, agents, or objects in your software?
- Are you [mapping what and how things](#) relate to each other?

## Interaction

- Are you [drawing a process](#) that includes tasks that occur in an order?
- Are you [drawing messages](#) between actors, agents, and objects?

- Are you drawing the flow of data through the system?

Then, [use a data flow diagram](#).

## Messages

- Is the relationship of things and messages key to capturing the subject?

Then, [use a communication diagram](#).

- Is the timing of messages shown on a timeline key?

Then, [use a sequence diagram](#).

## Software structure

- Is your diagram showing the layout of an interface?

Then, [use a wireframe](#).

- Is your diagram showing the composition of elements (such as modules of the software)?

Then, [use a package diagram](#).

- Is your diagram showing the relationship of database entities?

Then, [use an entity relationship diagram](#).

- Is your diagram showing the relationship of concepts, entities, or objects and it isn't a database?

Then, [use a class diagram](#).

- Is your diagram showing the relationship of hardware or software objects?

Then, [use a component diagram](#).

## Other chart types

Visual thinking and communication and thinking is an enormous subject. The focus in this guide is the most common diagram types that you are likely to use in capturing and communicating technical information. Some other visualizations and drawing types include:

- **Sketchnote.** Sketchnotes are visual notes created from a mix of handwriting, drawings, hand-drawn icons, stylized lettering, shapes, and visuals such as arrows, boxes, and lines. For more information, see [Roh Design](#).
- **Storyboard** A storyboard is a visual presentation of a sequence, typically for a film, that breaks down into single panels, or drawings. Each drawing may include notes and arrows for camera direction, dialogue, or in the case of interactive design, interaction. For more information, see [Studio Binder](#).
- **Social media graphs/Network graph**. A network graph is made of a collection of nodes (points) and edges (lines). For more information, see [Any Chart](#).
- **Mind map.** A mind map is a diagram used to organize information in a hierarchy. The map can be drawn by placing a single concept in the middle of your page, and then adding concepts and connecting them with a line. For more information, see [Mind map on Wikipedia](#).

## Next steps

Find a recipe and create your drawing [Diagram cookbook](#).

# How to draw a diagram from a diagram type

4/7/2021 • 2 minutes to read

You can draw a diagram even if you can't draw a realistic illustration freehand. Identify the type of diagram that will capture the concept you are trying to visualize, and either draw the diagram on a white board or use a Visio template.

The following process shows the steps to draw a diagram:

## Steps to draw a diagram

Select an existing diagram format depending on your subject. Each element in the diagram had a definition. For instance, in a class diagram, a class has a name, and a set of attributes and methods. You can build your diagram one element at a time. Add an element, and then a second element, and connect the two. Continue to add detail to your diagram until you have captured your information.

1. Select a diagram type. For guidance on choosing a diagram type, see [How to choose a diagram type](#).
2. Add more things, relationships, or groups. Each diagram type uses specific elements. An architectural diagram may use different pieces of hardware such as servers, routers, and storage devices. A class diagram uses classes connected by a range of relationships such as inheritance, aggregation, and composition.
3. Draw or review your diagram.
4. Review your diagram for completeness, missing information, and extra information. Number the elements of your diagram.
5. Finally give the diagram a title, make sure the diagram can be found and updated, and make notes to accompany your diagram.

After completing the draft of your diagram, you can [review it to refine the detail](#).

## Next steps

Review [Diagrams for technical communicators](#).

# How to draw a diagram from scratch

4/7/2021 • 2 minutes to read

You can draw a diagram even if you can't draw a realistic illustration *freehand*. A diagram of boxes and arrows is enough to capture most of the technical concepts that a writer may need to visualize.

The following process shows the steps to draw a diagram:

## Steps to draw a diagram

You can build your diagram one element at a time. Add a box, and then a second box, and connect the two. Continue to add detail to your diagram until you have captured your information.

1. Begin with a blank slate, a whiteboard, or screen.
2. Draw the first thing as a square. Label it.
3. Draw a second thing as a square and label it.
4. Connect the two things with a line. You have drawn two things and their relationship. Label the thing with their relationship.
5. Create a logical group for your items if needed.
6. Assess the entire diagram. Continue to add objects, relationships, and groups until you are satisfied with the completeness of your diagram. If you need to add more information, return to step 2. If you have enough information, then you can proceed.
7. Clear up stray items in your diagram.
8. Number the items in your diagram so that reviewers can specify the item when providing feedback.
9. Finally give the diagram a title, make sure the diagram can be found and updated, and make notes to accompany your diagram.

After completing the draft of your diagram, you can [review it to refine the detail](#).

## Next steps

Review [Diagrams for technical communicators](#).

# How to provide feedback to a diagram

4/7/2021 • 2 minutes to read

You can analyze a diagram by walking through the diagram and assessing what the diagram is communicating and if the diagram meets this purpose.

The following process shows the steps to analyze a diagram:

## Steps to analyze a diagram

The following checklist will help you review the context and elements to validate the purpose, completeness, and if there is extra detail in a diagram.

1. Review the diagram and note the following things:
  - a. Answer the who, what, where, when, why of the diagram.
    - a. Is it clear on the first glance what the diagram is trying to communicate?
    - b. Can you write down the purpose of the diagram in a single statement? If you need multiple statements, then write them down.
    - c. For each purpose, are there extra elements in the diagram that do not contribute to the purpose?
    - d. Does the diagram have over nine elements? Using a total of nine elements is commonly recognized as the greatest number of elements you would use to successfully communicate the details of the diagram. If the diagram has more than nine elements consider combining elements, using summary elements that can be expanded to show detail, or breaking your diagram into parts.
    - e. How might you simplify the diagram for each purpose?
    - f. Do you need to provide multiple views of the diagram so that each view has nine or fewer elements and focuses on a specific purpose?
  2. From the top-left side of the diagram, travel in rows from left to right and label each thing in the diagram with a number. There may already be labels on the diagram, but you are assessing the diagram for yourself.
  3. For each thing you have identified, draw a line, or label a line if one already exists, to other things for which the thing has a relationship. When you are done identifying relationships, travel from the top-left side to the bottom-right side and for each thing and give each relationship a unique letter, A-Z, AA-ZZ, etc.
  4. Are there groups or things that contain things in your diagram? Identify these. Unlike a relationship that is defined by one thing connected to another, these relationships are identified as one thing contained in another thing. This relationship may be identified as contains, child, etc. A container thing may have multiple sub-things.
  5. Create a table of your things. Each column of your table will be an attribute of your things. Do different things in your diagram? A unique thing may have distinct attributes.
  6. Create a table of your relationships. Each relationship will have the start with the following attributes: ID,

name, source, target, and may have additional attributes such as type or whatever you need. Do you have unique attributes?

7. When you are done, you will have at two master tables. You have within these tables, different sections for different things and different relationships. You will also have identified groupings and final views of the diagram.
8. Review the catalog of diagrams and determine which diagram forms would best communicate the purpose of your diagram. Choose a form, filter your data by the form, and follow the transformation/diagramming steps for the diagram type.

## Next steps

Review [Diagrams for technical communicators](#).

# Diagram cookbook

4/7/2021 • 26 minutes to read

The following diagrams are the most common diagram types you are likely to encounter when creating technical documentation. This is not a complete list, but does capture the major areas of illustrating a system, data, or set of concepts in relation to a technical topic.

For each diagram, you can find:

- The name of the diagram.
- The purpose of the diagram.
- A sample, hand drawn version of the diagram.
- Ingredients for the diagram. These are the things you need to have on hand to create the diagram. While you need something to start, the process of drawing the diagram may help you discover more things.
- The steps to draw the diagram on a whiteboard. All of these diagrams can be created by hand.
- How to find the Visio template and draw the diagram with the template.
- How to write the diagram using Mermaid, a common diagram as code language, that is fully supported with extension in Visual Studio Code and can be integrated into your GitHub workflow.

## Class diagram

Describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

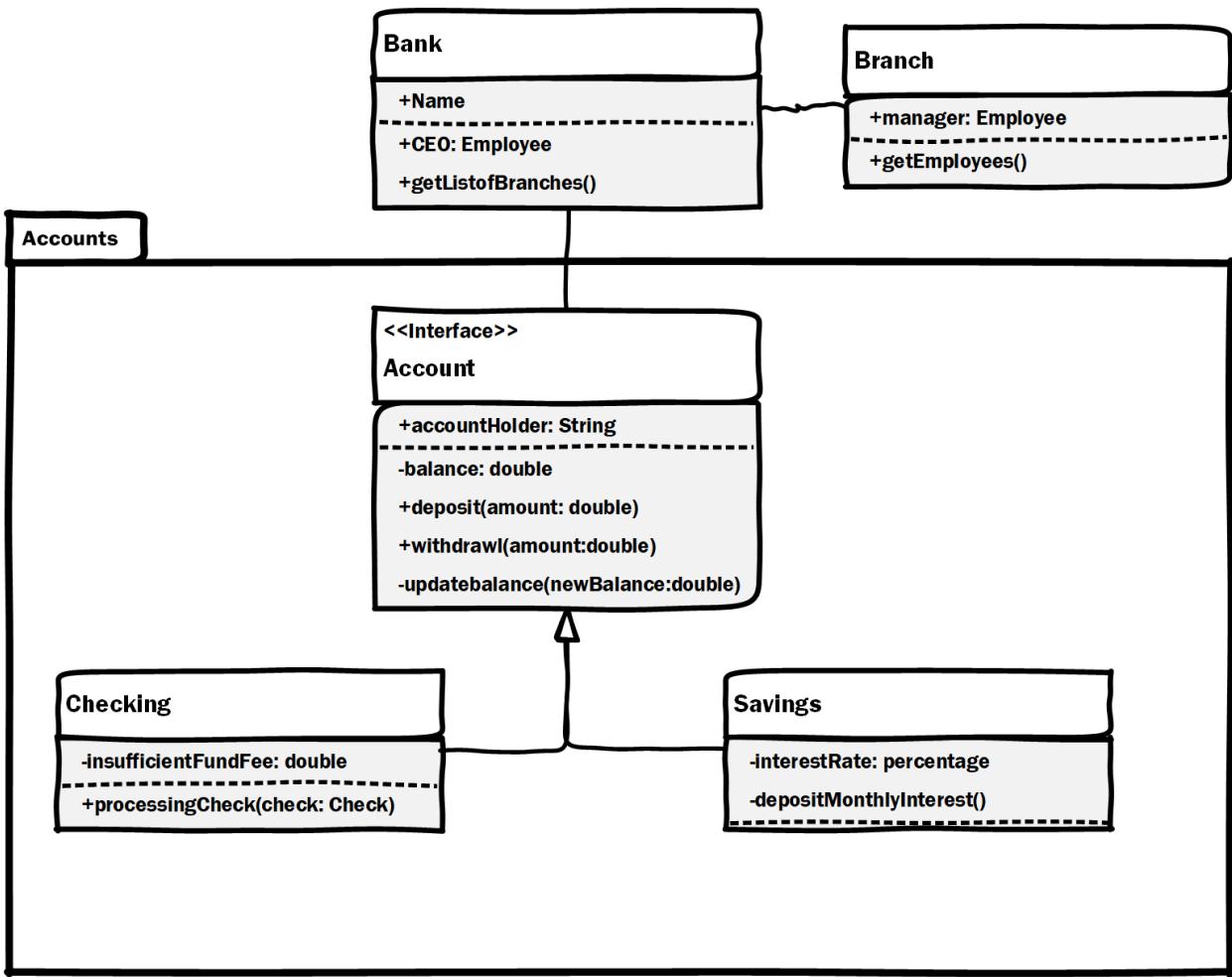
### Purpose

A class diagram shows objects and their relationships in what might be called an object model. The class diagram can be used to describe classes in object-oriented programming, or in a formal description of an area of knowledge the relationships of concepts.

### Type

UML, structural

### Sample



## Ingredients

The following table provides the ingredients for this type of diagram.

INGREDIENT TYPE	DESCRIPTION
Package	Represents a package in a process.
Object	<ul style="list-style-type: none"> <li><b>Attribute</b> - An attribute is a property of an object. It may be represented by a data type.</li> <li><b>Operations</b> - An operation is an action that the class may be able to take. It is often referred to as a method.</li> <li><b>Interface</b> - Specifies the externally visible operations of a class, component, package, or other element without specifying internal structure.</li> <li><b>Enumeration</b> - Describes a data type consisting of a set of named values.</li> </ul>

INGREDIENT TYPE	DESCRIPTION
Relationship	<ul style="list-style-type: none"> <li>• <b>Dependency</b> - One object depends on another.</li> <li>• <b>Association</b> - This is a generic relationship between two objects.</li> <li>• <b>Aggregation</b> - One object is contained in another; however the object will remain if the container is removed.</li> <li>• <b>Composition</b> - One object is contained in another; when the container is removed, the contained object is removed.</li> <li>• <b>Inheritance</b> - One object acquires the attributes and operations of another class. The source class is a superclass and the class inheriting properties is a subclass.</li> <li>• <b>Implementation</b> - One class realizes the expression of another class. For instance, a document is an instance of a template.</li> </ul>
Multiplicity	An attribute of an association between classes multiplicity indicates the cardinality in both target and source, and source and target.

### Steps (recipe)

1. Draw an object as a rectangle.
2. Divide the rectangle into two sections.
3. In the upper half list, the attributes. These are things that make up the object.
4. In the lower half list, the things the object can do.
5. Draw a second or more object and define them.
6. Connect one object to another object and determine the relationship.
7. If the two objects have an association, do they have multiplicity? Note the multiplicity.
8. Continue to add objects and relations until you have a complete model.

### Steps to find the Visio

1. Open Visio and select New.
2. Type UML Class in the Search box.
3. Open the UML class template and select Create.
4. For instructions see, Create a UML class diagram.

### Example in mermaid

```

classDiagram
    Animal <|-- Duck
    Animal <|-- Fish
    Animal <|-- Zebra
    Animal : +int age
    Animal : +String gender
    Animal: +isMammal()
    Animal: +mate()
    class Duck{
        +String beakColor
        +swim()
        +quack()
    }
    class Fish{
        -int sizeInFeet
        -canEat()
    }
    class Zebra{
        +bool is_wild
        +run()
    }
}

```

For more information about drawing a class diagram in Mermaid, see: <https://mermaid-js.github.io/mermaid/diagrams-and-syntax-and-examples/classDiagram.html>

### Additional information

- **Notes** - Class diagrams are useful in object orientated programming and also in modeling domains or conceptual areas. A useful text about object modeling is the book Domain Driven Design by Eric J. Evans available via the Microsoft O'Reilly's Site License.
- **Agile Modeling** - <http://agilemodeling.com/artifacts/classDiagram.htm>
- **Wikipedia** - [https://en.wikipedia.org/wiki/Class\\_diagram](https://en.wikipedia.org/wiki/Class_diagram)
- **UML** - <https://www.uml-diagrams.org/class-diagrams-overview.html>
- **Microsoft support** - <https://support.microsoft.com/office/create-a-uml-class-diagram-de6be927-8a7b-4a79-ae63-90da8f1a8a6b?ns=visio365&version=90>

## Communication diagram

Models the interactions between objects or parts in terms of sequenced messages. Communication diagrams represent a combination of information taken from Class, Sequence, and Use Case Diagrams describing both the static structure and dynamic behavior of a system.

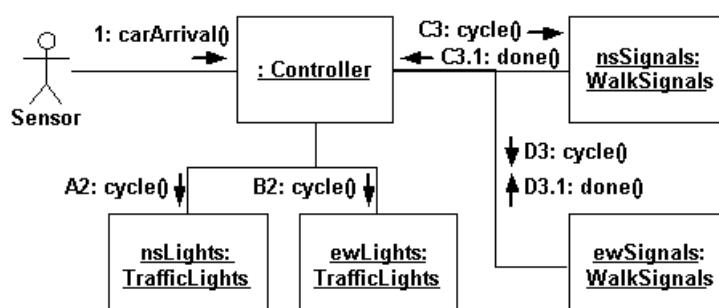
### Purpose

Depicts message patterns and message contents between elements of your system.

### Type

UML, behavioral

### Sample



## Ingredients

The following table provides the ingredients for this type of diagram.

INGREDIENT TYPE	DESCRIPTION
Actor	The actor represents the role of a user or external system. Lifeline. The lifeline is a named element in the system. In this diagram, the lifeline is the entity beginning with a colon.
Message	Is a communication between shapes in the diagram. A message has a source (sender) and target (receiver). The message is labeled with the vehicle of the message.
Diagram Overview	This is a container to mark the boundaries of your system and include any additional contextual notes about your system.

## Steps (recipe)

1. Collect the various objects in your system that are connected by communication.
2. Consider how each object in the system communicates with other objects.
3. Draw each object as a rectangle and label the object :object-name.
4. Draw an arrow from object sender (source) to each object receiver (target). Label the arrow with the vehicle that carries the message.

## Steps to find the Visio

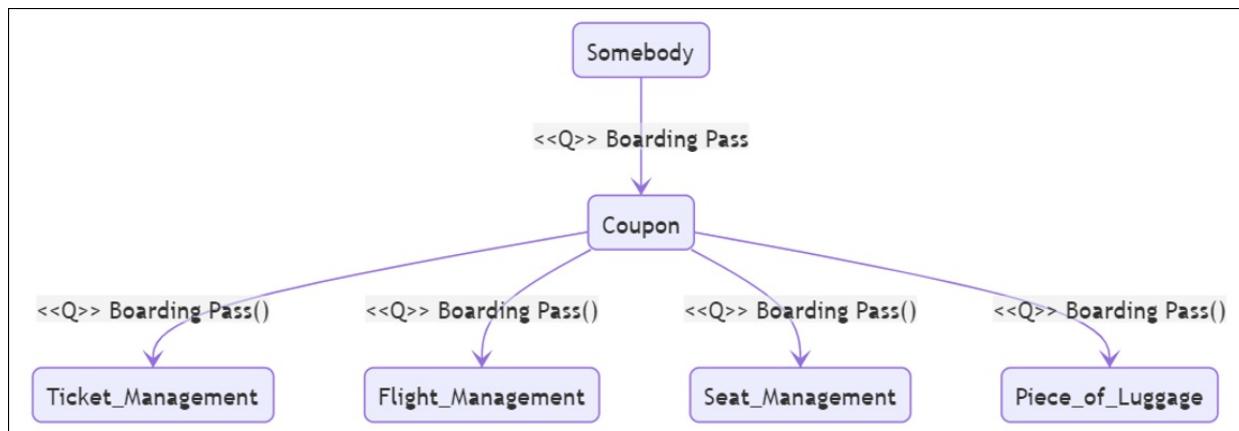
1. Open Visio. Select New and then type UML communication.
2. Select the blank template and select Create.
3. Drag the shapes from the Shapes pane that capture your system.
4. Connect the messages in your system.
5. You can wrap your diagram in the Diagram Overview.

For the Visio instructions see: <https://support.microsoft.com/office/create-a-uml-communication-diagram-911956f4-5f19-4a58-97a3-bb14110a5ed1>

## Example in mermaid

Mermaid doesn't have a specific diagram type. You can use the state diagram.

```
stateDiagram-v2 Somebody --> Coupon : <'> Boarding Pass
Coupon --> Ticket_Management : <'> Boarding Pass()
Coupon --> Flight_Management : <'> Boarding Pass()
Coupon --> Seat_Management : <'> Boarding Pass()
Coupon --> Piece_of_Luggage : <'> Boarding Pass()""""
```



- Agile Modeling - <http://agilemodeling.com/artifacts/communicationDiagram.htm>

- Wikipedia - [https://en.wikipedia.org/wiki/Communication\\_diagram](https://en.wikipedia.org/wiki/Communication_diagram)
- UML - <https://www.uml-diagrams.org/communication-diagrams.html>

<!-- ----- -->

## Component diagram

Depicts how components are wired together to form larger components or software systems.

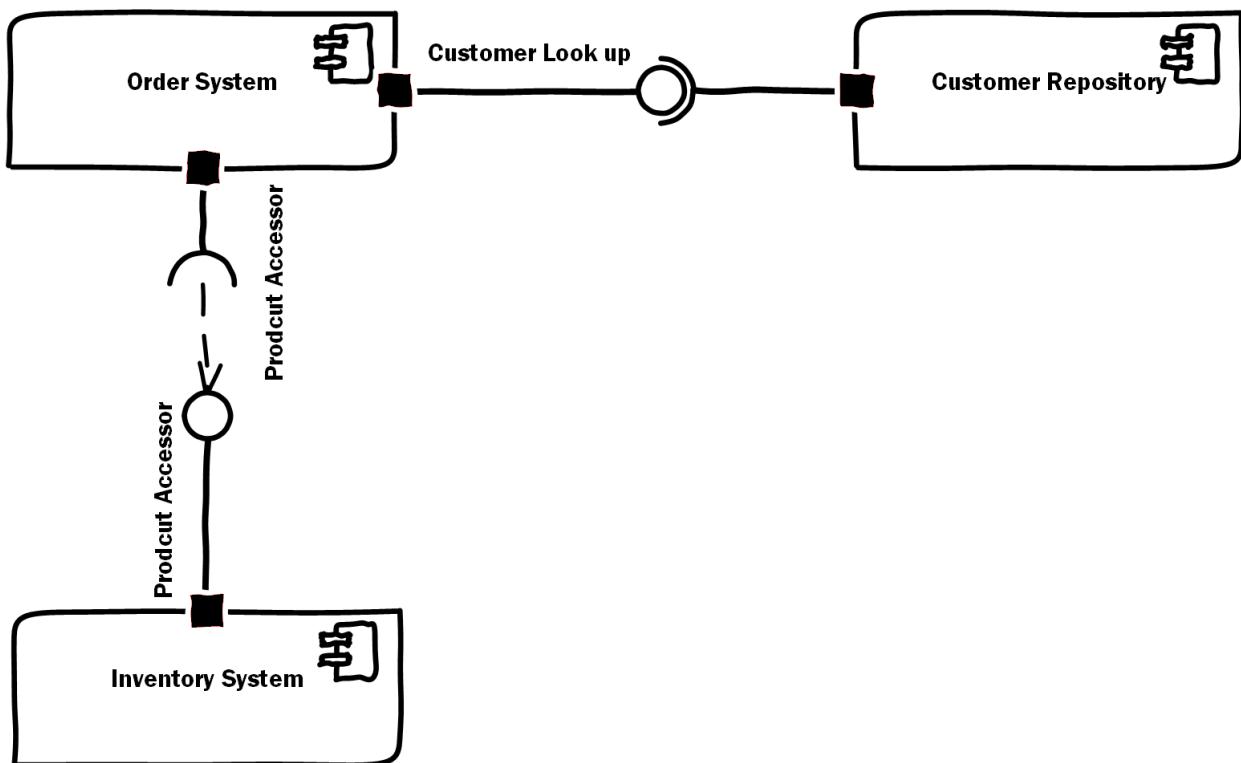
### Purpose

Depicts how components are wired together to form larger components or software systems.

### Type

UML, behavioral

### Sample



### Ingredients

- **Component** - Use component shapes for each functional unit in your system or application.
  - Package
  - Association
  - Directed association
  - Composition
  - Dependency
  - Aggregation
- **Required interface** - Use the Required Interface when you want to specify a dependency on a class/interface.
- **Provided interface** - Use the Provided Interface shape when you want to specify the realization of a class/interface.

### Steps (recipe)

1. Collect the various modules in your system.
2. Consider how the relationships or associations of each module in the system.

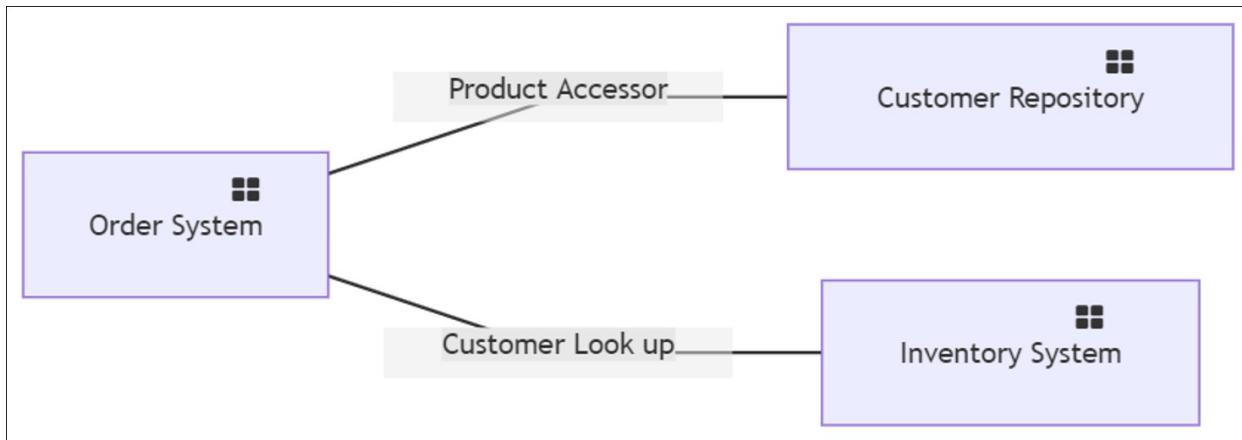
3. Draw each module as a component. Note some functional elements contain subelements (or submodules). In the diagram, you can place a component within a component.
  4. Add the associations and interfaces.

## Steps to find the Visio

1. Open Visio. Select New and then type UML component.
  2. Select the blank template and select Create.
  3. Add your components for your system.
  4. Add the association between components.
  5. Add the interfaces to your system.

## Example in mermaid

Mermaid doesn't have a specific diagram type. You can use the graph diagram.



## **Additional information**

- Agile Modeling - <http://agilemodeling.com/artifacts/componentDiagram.htm>
  - Wikipedia - [https://en.wikipedia.org/wiki/Component\\_diagram](https://en.wikipedia.org/wiki/Component_diagram)

# Data flow diagram

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

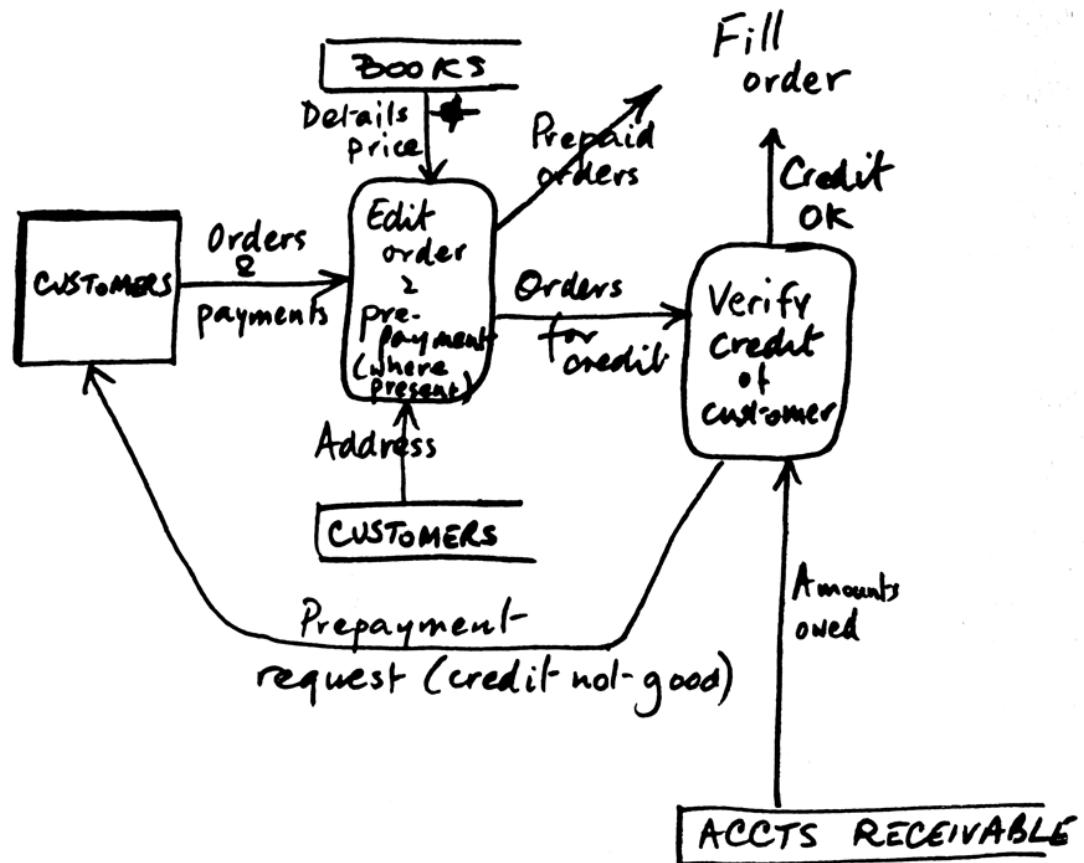
## Purpose

Represents a flow of data through a process or a system (usually an information system).

## Type

## System, structural

## Sample



## Ingredients

- **Process or interface** - Specifies the object being acted upon.
- **Data store** - A location where data is at rest.
- **Data flow** - A directed arrow.

## Steps (recipe)

1. Collect the various elements of your system.
2. Draw each interface, each process, and each data store in the system.
3. Walk the flow of data from outside of the system and into the system and from inside of the system to the outside of the system.

## Steps to find the Visio

1. Open Visio. Select New and then type data flow model.
2. Select the blank template and select Create.
3. Add the elements of your system.
4. Add the data stores
5. Add the interfaces
6. Add the processes.
7. Add the data flow that connects interfaces, processes, and data stores.

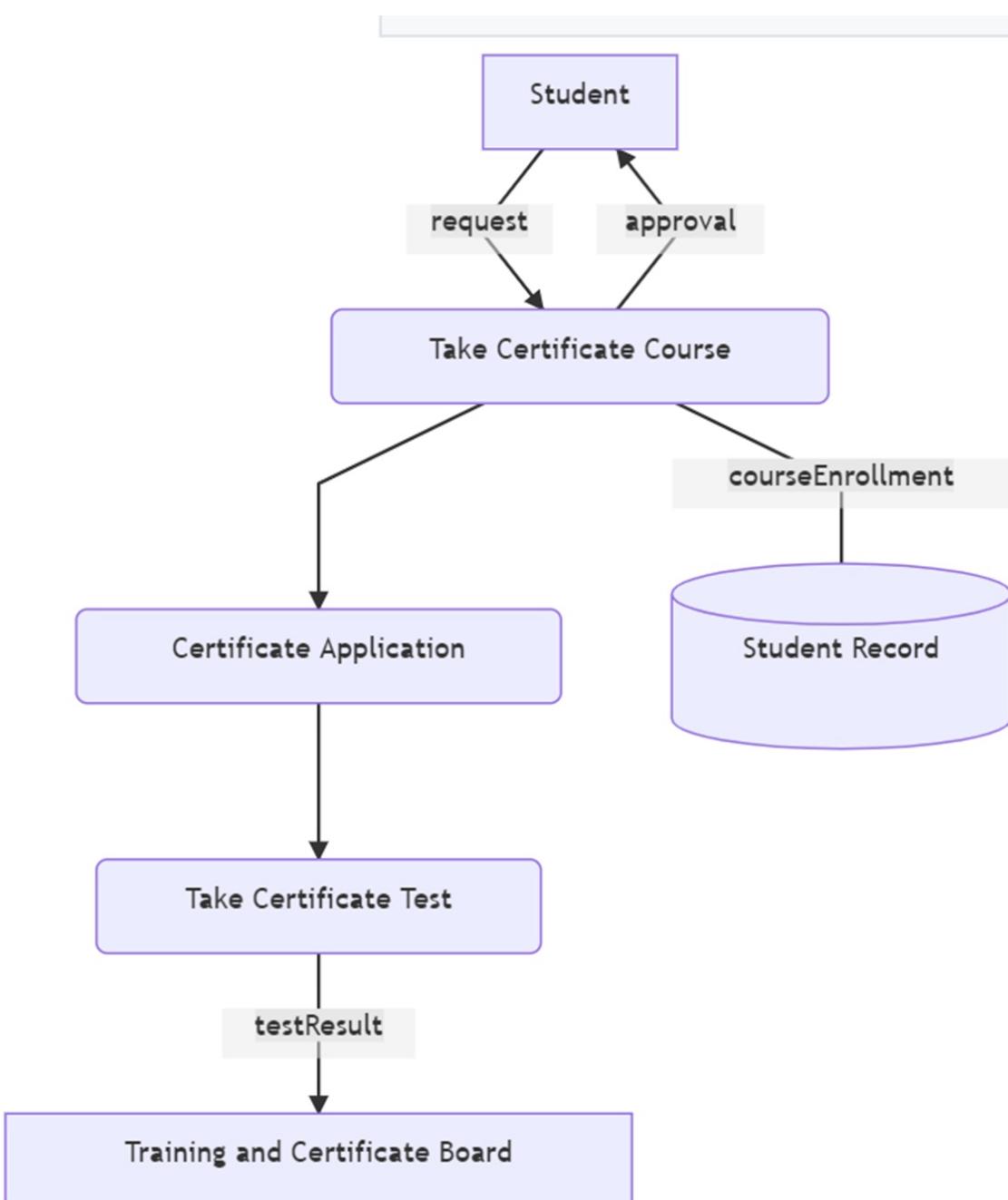
For the Visio instructions see: <https://support.microsoft.com/office/create-a-gane-sarson-data-flow-model-diagram-0d7ca68a-0cd2-4022-8cc9-4003bbae3724>

## Example in mermaid

Mermaid doesn't have a specific diagram type. You can use the graph diagram.

Process: a node with rounded edges: ( Node ) Interface: a node with square edges: [ Node ] Data Store: a node with a bucket shape: [( Node )]

```
graph TB
    A[Student]
    B(Take Certificate Course)
    C(Certificate Application)
    D(Take Certificate Test)
    E[(Student Record)]
    F[Training and Certificate Board]
    A-- request -->B
    B-- approval -->A
    B --> C
    C --> D
    B-- courseEnrollment ---E
    D-- testResult -->F
```



## Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/dataFlowDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Data-flow\\_diagram](https://en.wikipedia.org/wiki/Data-flow_diagram)
- UML - <https://www.uml-diagrams.org/information-flow-diagrams.html>

## Deployment diagram

A deployment diagram is a structure diagram, which shows the architecture of a system as deployment (distribution) of software artifacts to deployment targets. Artifacts represent concrete elements in the physical world that are the result of a development process. Examples of artifacts are executable files, libraries, archives, database schemas, configuration files, etc.

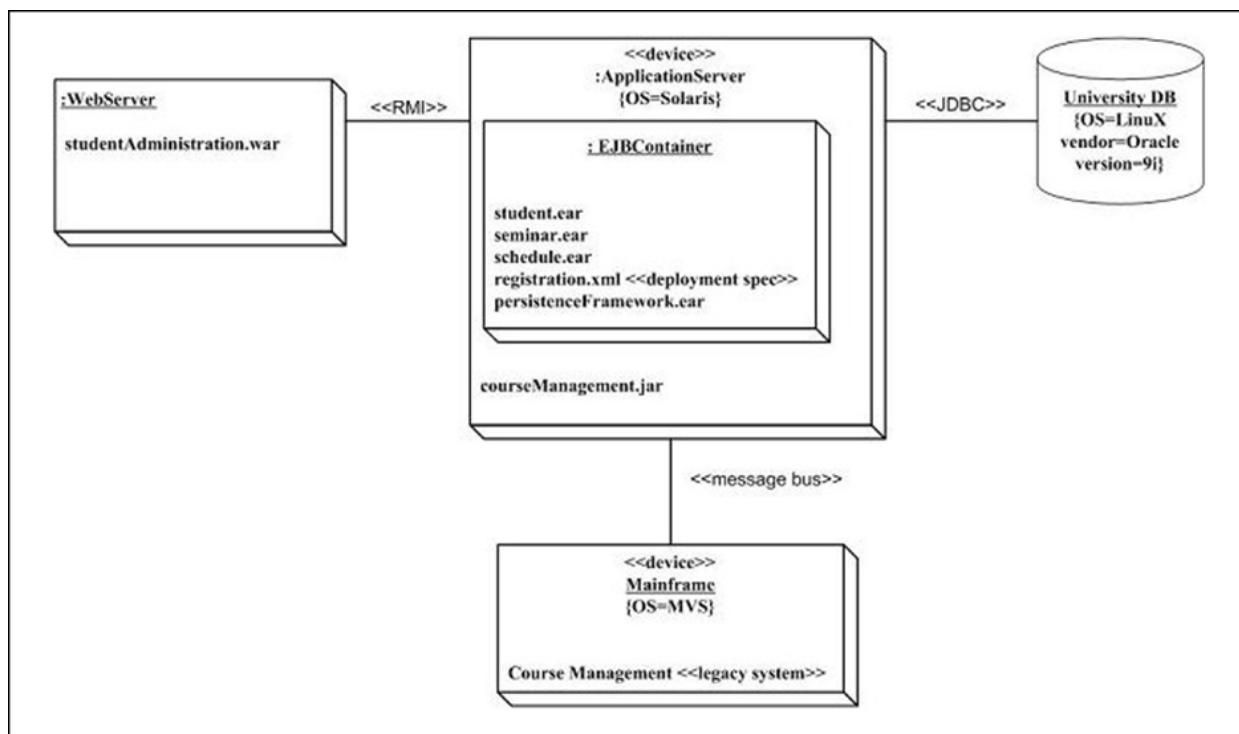
### Purpose

A deployment diagram depicts a static view of the run-time configuration of processing nodes and the components that run on those nodes.

### Type

UML, structural

### Sample



### Ingredients

A deployment target is represented by nodes, which are either **hardware devices** or some **software execution environment**. Nodes can be connected through communication paths to create networked systems of arbitrary complexity. Nodes can contain other nodes or software artifacts.

### Steps (recipe)

1. Identify the scope of the model that will be used to create the diagram.
2. Identify the distribution architecture. Draw the nodes that represent this architecture, such as the hardware devices or software execution environment.
3. Draw the database if it is represented in your environment.
4. Draw the connections between the different nodes. This step helps to show the
5. Add addition detail to each node to show supporting infrastructure, such as supporting files or software that is deployed on each node, critical information for anyone involved in development, installation, or operation

of the system.

## Steps to find the Visio

1. Open Visio and select New.
2. Enter **UML Deployment** in the Search box.
3. Select the template that best suits your design, then click **Create** to modify the template diagram.
4. For instructions see, [Create a UML deployment diagram](#).

## Example in mermaid

For information about drawing a diagram in Mermaid, see: <https://mermaid-js.github.io/mermaid/#/n00b-gettingStarted>

## Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/deploymentDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Deployment\\_diagram](https://en.wikipedia.org/wiki/Deployment_diagram)
- UML - <https://www.uml-diagrams.org/deployment-diagrams-overview.html>

## Entity relationship diagram

An entity relationship model, describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between entities (instances of those entity types). In software engineering, an ER model is commonly formed to represent things a business needs to remember in order to perform business processes. Consequently, the ER model becomes an abstract **data model**, that defines a data or information structure, which can be implemented in a database, typically a relational database.

There is a tradition for ER/data models to be built at two or three levels of abstraction:

- Conceptual data model
- Logical data model
- Physical data model

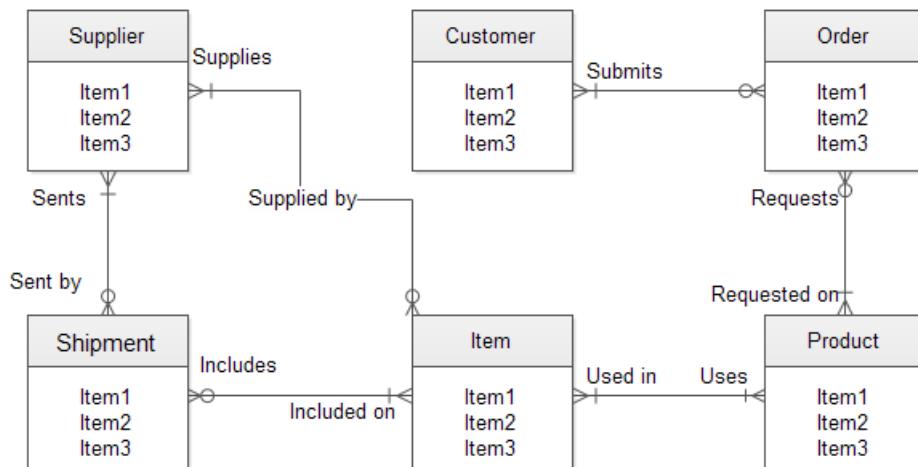
## Purpose

An entity relationship model (or ER model) describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types (which classify the things of interest) and specifies relationships that can exist between **entities** (instances of those entity types).

## Type

System, structural

## Sample



## Ingredients

There are three primary parts of an ERD:

- **Entities** - The tables in your database. Entities are nouns. Common classifications are concepts, locations, roles, events, and things.
- **Attributes** - Facts that describe each table. They are nouns, and they often become the columns of tables.
- **Relationships** - Relationships are verbs. They indicate the associations between entities.

## Steps (recipe)

1. Identify the **entities**. The first step in making an ERD is to identify all of the **entities** you will use.
2. Identify **relationships**. Look at two **entities**, are they related?
3. Describe the **relationship**. How are the **entities** related?
4. Add attributes.
5. Complete the **diagram**.

## Steps to find the Visio

1. Open Visio and select New.
2. Enter Entity in the Search box.
3. Select the template that best suits your design, then click Create to modify the template diagram.
- 4.

For instructions see, [Create entity relationship diagrams in Visio](#).

## Example in mermaid

For information about drawing a diagram in Mermaid, see: <https://mermaid-js.github.io/mermaid/#/entityRelationshipDiagram?id=entity-relationship-diagrams>

## Additional information

- **Agile Modeling** - <http://agiledata.org/essays/agileDataModeling.html#InitialDomainModel>
- **Wikipedia** - [https://en.wikipedia.org/wiki/Entity%20relationship\\_model](https://en.wikipedia.org/wiki/Entity%20relationship_model)
- **UML** - [Entity Relationship Modeling with UML](#)

# Network diagram

Network diagrams are commonly used to depict hardware nodes and the connections between them. Network diagrams are arguably a high-level form of UML deployment diagram with extensive use of visual stereotypes.

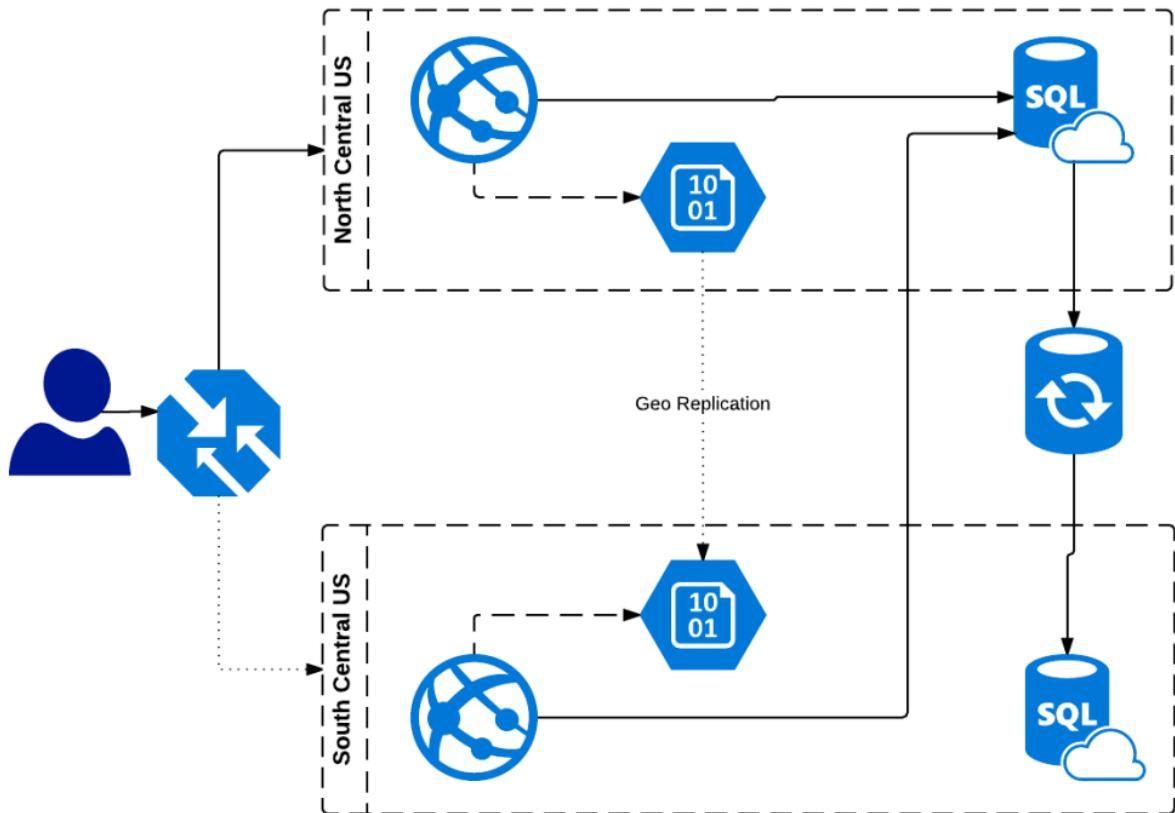
## Purpose

Use a deployment diagram to show the structure of the run-time system and communicate how the hardware and software elements that make up an application will be configured and deployed.

## Type

System, structural

## Sample



## Ingredients

The following table provides the ingredients for this type of diagram.

- **Node instance** - Use Node Instance shapes when you want to specify an instance of a run-time computational or physical device. Node instances and artifact instances by diagramed using visual stereotypes for devices such as network gear, routers, or physical firewalls, and computational instances such as databases, web servers or file servers.
- **Artifact instance** - Put Artifact Instance shapes inside node instance shapes to deploy artifacts.
- **Hierarchical nodes** - Nest node shapes inside each other when you want to show them hierarchically.
- **Package nodes** - Use Package shapes when you need to represent containing elements like a folder.
- **Diagram Overview** - This is a container to mark the boundaries of your system and include any additional contextual notes about your system.
- **Connectors** - Connectors connects diagram source and target of communication channels between node instances.

### TIP

You can find Azure icons at: <https://cds-icons.azurewebsites.net/>

## Steps (recipe)

1. Collect the various objects in your system that are connected by communication paths.
2. Consider how each object in the system communicates with other objects.
3. Draw each object as a rectangle and label the object.
4. For each object, consider its composition of subsystems.and draw these elements with in the container object.
5. Draw an arrow from object sender (source) to each object receiver (target). Label the arrow with the vehicle that carries the message

## Steps to find the Visio

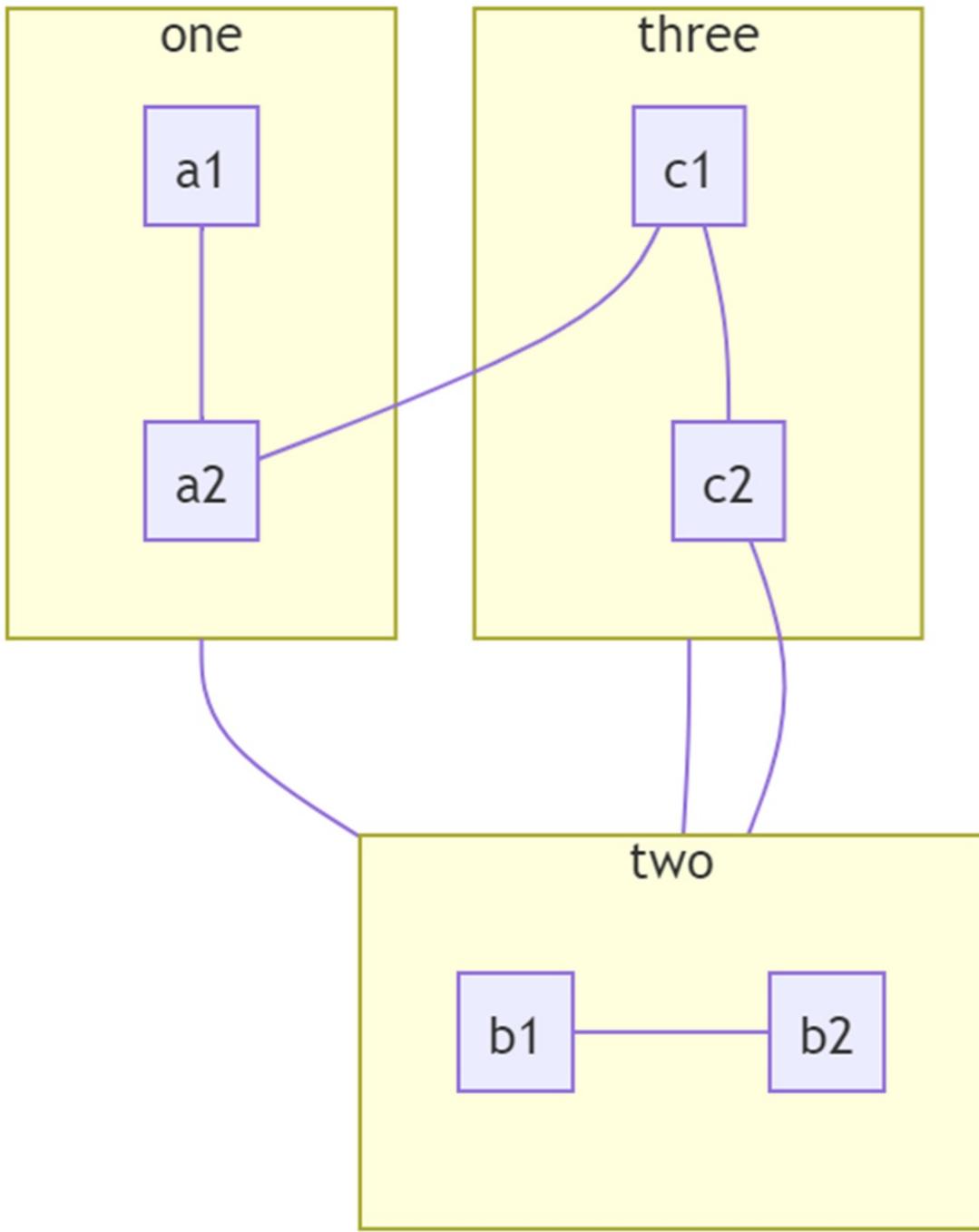
1. Open Visio. Select New and then type UML deployment.
2. Select the blank template and select Create.
3. Add your first object (a computational or physical device) and then continue to draw additional objects.
4. For each object, add artifacts as appropriate inside of each node.
5. Connect objects along communication channels.
6. Optionally, you may add hierarchical nodes that show larger systems, distribution of nodes across a hybrid cloud or resource groups or other physical or logical divisions.
7. Optionally, you can contain your diagram in a diagram overview object.
8. When you are done with your UML deployment diagram, you can replace each node instance or artifact instance with an icon that is appropriate to the instance type.

For instructions:

<https://support.microsoft.com/en-us/office/create-a-uml-deployment-diagram-ef282f3e-49a5-48f5-a6ae-69a6982a4543?ns=visio365&version=90&ui=en-us&rs=en-us&ad=us>

### Example in mermaid

```
flowchart TB
    c1---a2
    subgraph one
    a1---a2
    end
    subgraph two
    b1---b2
    end
    subgraph three
    c1---c2
    end
    one---two
    three---two
    two---c2
```



#### Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/networkDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Graph\\_drawing](https://en.wikipedia.org/wiki/Graph_drawing)

## Package diagram

A package diagram in the Unified Modeling Language depicts the organization and dependencies between various elements (packages) that make up a model. A package groups together elements that are semantically related and might change together. It is a general purpose mechanism to organize elements into groups to provide better structure for a system model.

#### Purpose

Package diagrams can use packages that contain and organize either **classes**, **data entities**, or **use cases** of a software system. Packages can represent the different layers of a software system to illustrate the layered architecture of a software system. The dependencies between these packages can be adorned with labels / stereotypes to indicate the communication mechanism between the layers.

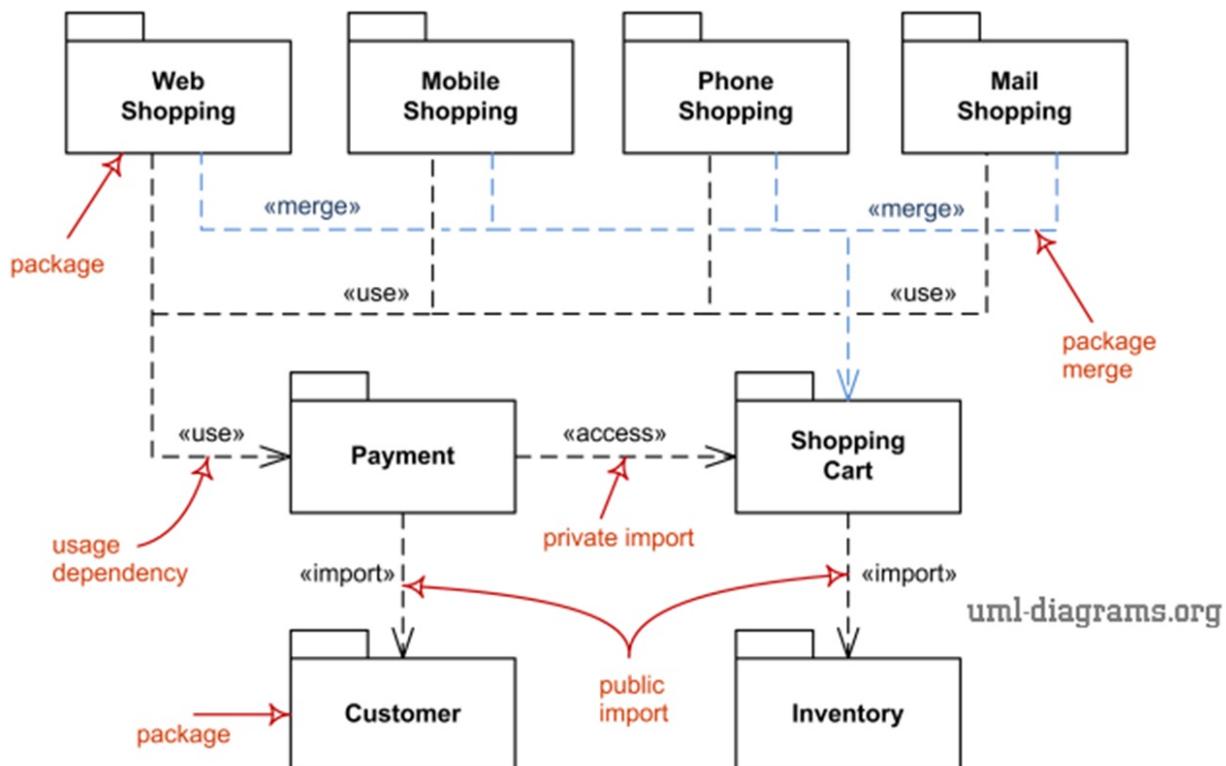
Package diagrams can be useful to do the following:

- Create an overview of a large set of model elements
- Organize a large model
- Group related elements
- Separate namespaces

### Type

UML, structural

### Sample



### Ingredients

The following elements are typically drawn in a package diagram:

- package
- packageable element
- dependency
- element import
- package import
- package merge

### Steps (recipe)

To create a Package Diagram:

1. Select **Diagram > New** from the toolbar.
2. In the **New Diagram** window, select **Package Diagram** and click **Next**.
3. Enter **Racing Game Packages** as diagram name and click **OK** to confirm.
4. Click the **Package** button in **diagram** tool bar, then click on the blank area of the **diagram** to create the package.

### Steps to find the Visio

1. Open Visio and select **New**.

2. Under Template Categories, click Software, and then click UML Model Diagram.

3. Click Create.

For instructions, see [Create a UML package diagram](#).

### Example in mermaid

```
package Sales {  
    class Product {  
        +int id  
    }  
    Product o-- Employees.Manager  
}  
package Employees {  
    class Manager {  
        +int id  
    }  
}
```

### Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/packageDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Package\\_diagram](https://en.wikipedia.org/wiki/Package_diagram)
- UML - <https://www.uml-diagrams.org/package-diagrams-overview.html>

## Tree diagram

A tree structure or tree diagram is a way of representing the hierarchical nature of a structure in a graphical form. It is named a "tree structure" because the classic representation resembles a tree, even though the chart is upside down compared to a biological tree, with the "stem" at the top and the "leaves" at the bottom.

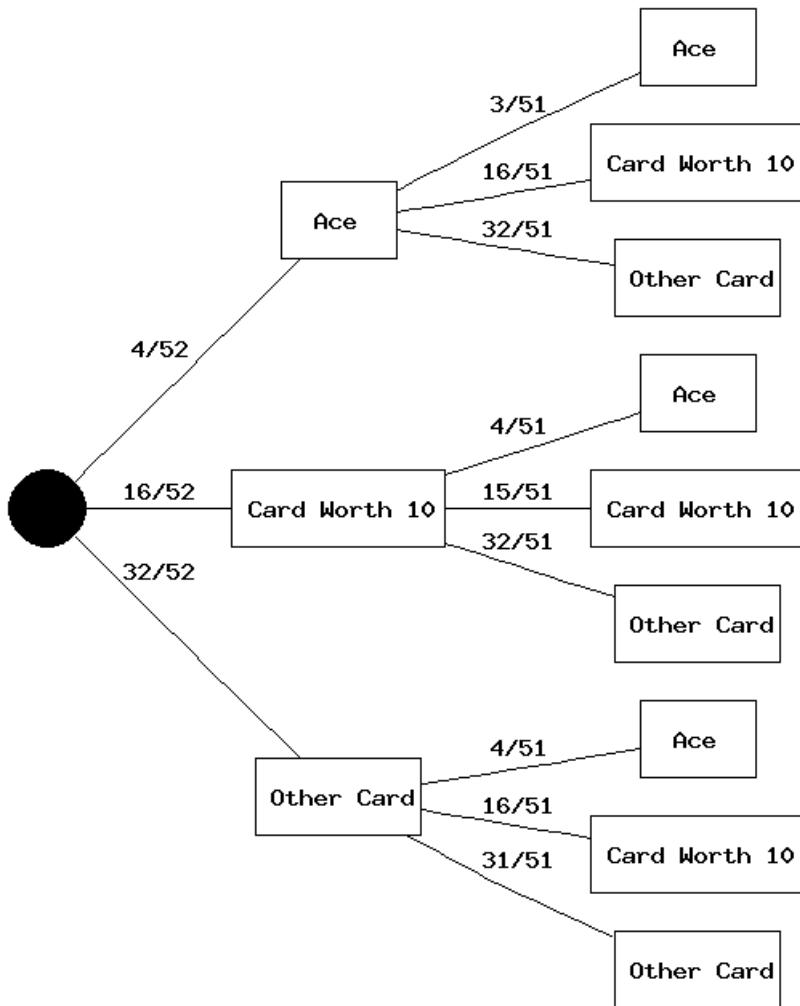
### Purpose

A tree diagram is simply a way of representing a sequence of events or the hierarchy of related objects.

### Type

Logical, structural

### Sample



## Ingredients

The tree elements are called "nodes". The lines connecting elements are called "branches". Nodes without children are called **leaf nodes**, "end-nodes", or "leaves". Every **finite** tree structure has a member that has no **superior**. This member is called the "root" or **root node**. The root is the starting node. But the converse is not true: infinite tree structures may or may not have a root node.

## Steps (recipe)

1. Create the top-level node (root).
2. Create subnodes (elements) below the related higher-level node(s).
3. Draw lines to connect the nodes.

## Steps to find the Visio

1. Click File > New > Templates > General, and then open **Block Diagram**.
2. From the **Blocks and Blocks Raised** stencils, drag block shapes onto the drawing page to represent stages in a tree structure.
3. To add text to a shape, select the shape, and then type.
4. Indicate relationships between the blocks by connecting the shapes.
5. Use tree shapes to represent hierarchical stages in a tree diagram.

For instructions, see [Create tree diagram in Visio](#).

## Example in mermaid

### Additional information

- Wikipedia - [https://en.wikipedia.org/wiki/Tree\\_structure](https://en.wikipedia.org/wiki/Tree_structure)

# Venn diagram

Describes

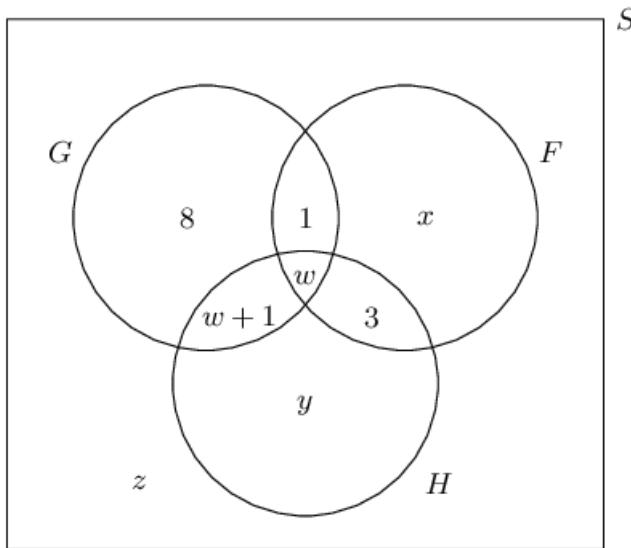
## Purpose

A Venn diagram is an illustration that uses circles to show the relationships among things or finite groups of things. Circles that overlap have a commonality while circles that do not overlap do not share those traits.

## Type

Logical, structural

## Sample



## Ingredients

These diagrams depict elements as points in the plane, and sets as regions inside closed curves. A Venn diagram consists of multiple overlapping closed curves, usually circles, each representing a set.

## Steps (recipe)

1. Create a circular shaped region that depicts a finite group.
2. Create a second circular shaped region that depicts another finite group, but draw the second region over part of the first region to represent the number of similar characteristics between the two groups. These overlaps show the relationship between the two groups.
3. Draw additional circular shaped regions to show additional relationships and their overlapping characteristics to the other regions.

## Steps to find the Visio

1. In Visio, on the File menu, click New > Business, and then click Marketing Charts and Diagrams.
2. Choose between Metric Units or US Units, and click Create.
3. From Marketing Diagrams, drag the Venn diagram shape onto the page.
4. Click a segment to select.
5. Click Fill in the Shape Styles area, and select a color.

## Example in mermaid

## Additional information

- Wikipedia - [https://en.wikipedia.org/wiki/Venn\\_diagram](https://en.wikipedia.org/wiki/Venn_diagram)

# Wireframe diagram

A wireframe, also known as a page schematic or screen blueprint, is a visual guide that represents the skeletal

framework of a graphic user interface. The wireframe depicts the page layout or arrangement of the website's content, including interface elements and navigational systems, and how they work together. The wireframe usually lacks typographic style, color, or graphics, since the main focus lies in functionality, behavior, and priority of content. In other words, it focuses on what a screen does, not what it looks like. Wireframes can be pencil drawings or sketches on a whiteboard, or they can be produced by means of a broad array of free or commercial software applications. Wireframes are generally created by business analysts, user experience designers, developers, visual designers, and by those with expertise in interaction design, information architecture and user research.

Wireframes focus on:

- The range of functions available.
- The relative priorities of the information and functions.
- The rules for displaying certain kinds of information.
- The effect of different scenarios on the display.

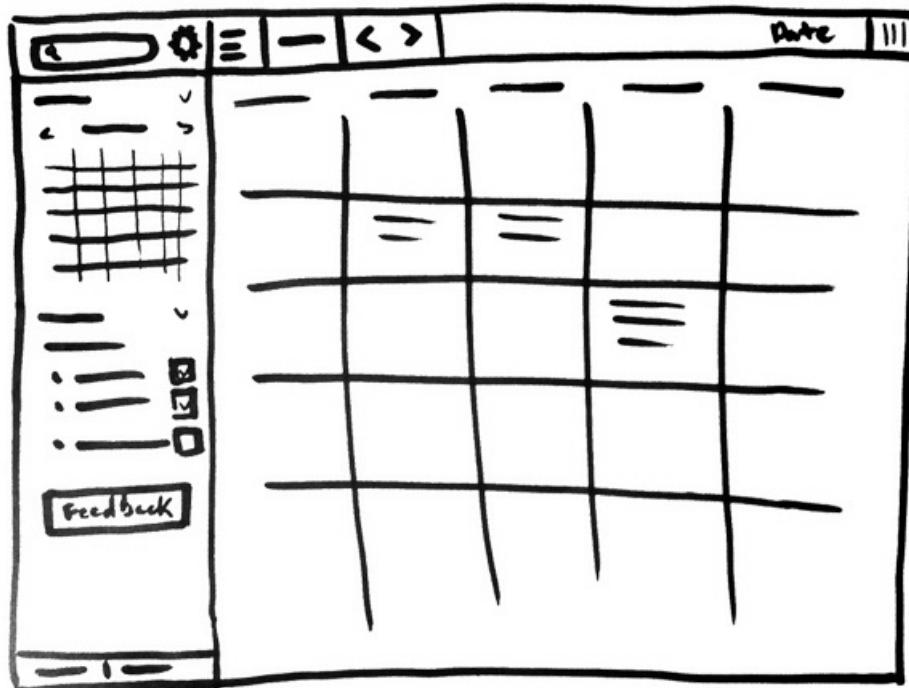
### Purpose

Wireframes are created for the purpose of arranging elements to best accomplish a particular purpose.

### Type

System, structural

### Sample



### Ingredients

The website wireframe connects the underlying conceptual structure, or [information architecture](#), to the surface, or [visual design](#) of the website. Wireframes help establish functionality and the relationships between different screen templates of a website.

### Steps (recipe)

1. Determine the shapes that will be represented on your web page.
2. Use the shapes to fill in the UI and show how the web page will function.

### Steps to find the Visio

1. Start by clicking the Dialogs stencil and dragging an Application form or a Dialog form onto the page.

2. Add other shapes from this stencil to outline the basic structure of the UI.
3. Click the Toolbars stencil to add menus and other application icons.
4. Use the shapes on the other four default stencils (Controls, Cursors, Common Icons, and Web and Media Icons) to fill in the UI and show the application's functionality. For instructions see, [Create a wireframe diagram for user interfaces](#).

#### **Example in mermaid**

#### **Additional information**

- **Related links** - <https://startupmusings.wordpress.com/2012/01/08/product-planning-series-information-architecture-flowcharts-and-wireframes/>
- **Agile Modeling** - <http://agilemodeling.com/artifacts/essentialUI.htm>
- **Wikipedia** - [https://en.wikipedia.org/wiki/Website\\_wireframe](https://en.wikipedia.org/wiki/Website_wireframe)

## **Activity diagram (flowchart)**

Activity diagrams are graphical workflow representations of stepwise activities and actions with support for choice, iteration, and concurrency. In the Unified Modeling Language, activity diagrams are intended to model both computational and organizational processes, as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores. This type of diagram describes the objects used, consumed, or produced by an activity and the relationship between the different activities.

A process flow diagram (PFD) is a type of activity diagram. PFDs show a way of representing the path through a process or a system. This type of diagram is a step-by-step process. The PFD displays the relationship between major steps in a process, it does not show minor details.

#### **Purpose**

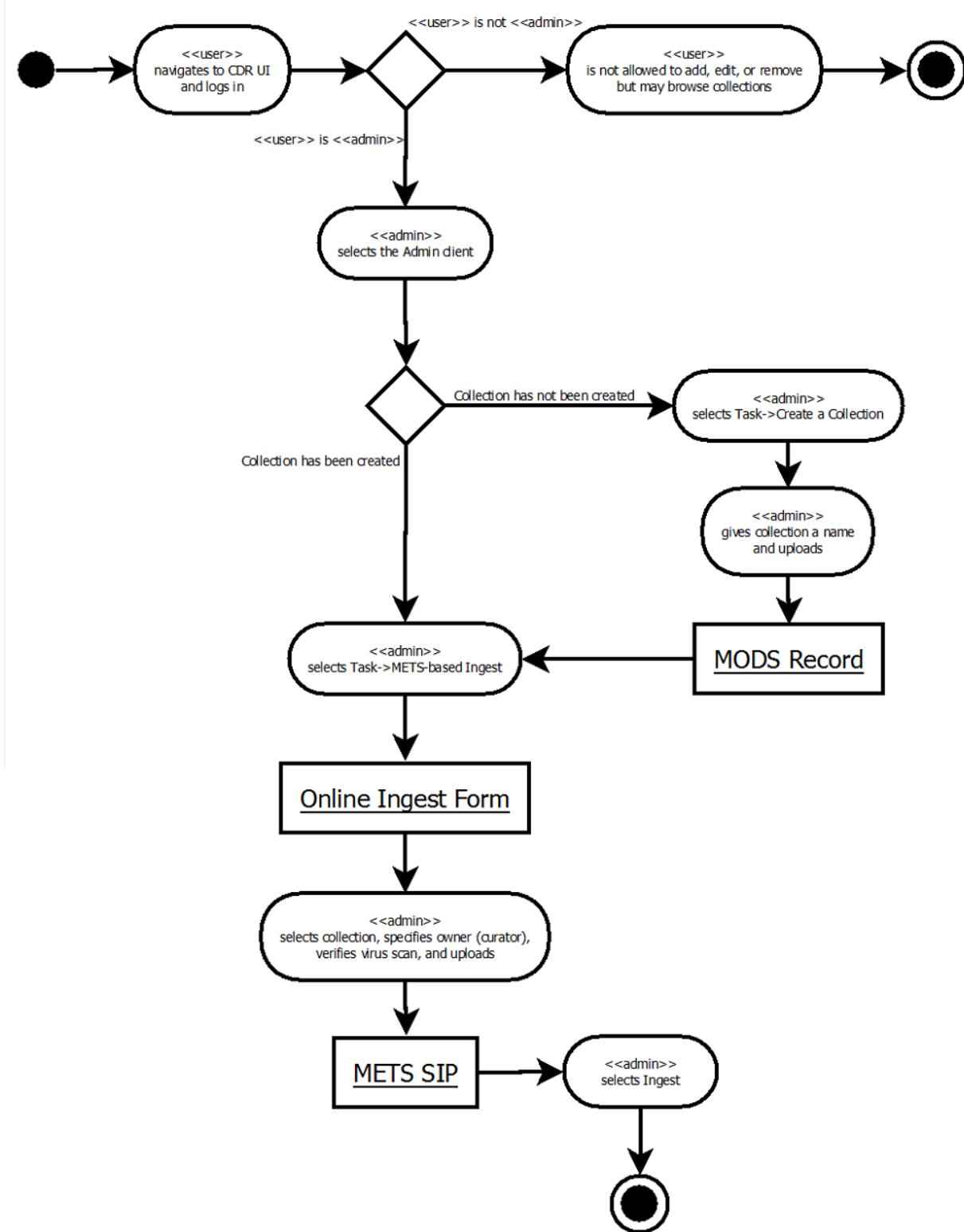
An activity diagram shows objects and choices representing the workflow of stepwise activities. This type of diagram is used to show an activity process from start to finish. This type of diagram displays the relationship between major steps in a process, it does not show minor details.

#### **Type**

UML, behavioral

#### **Sample**

## Ingest Procedure: Uploading a SIP Using the CDR UI



### Ingredients

An activity diagram contains activity nodes, which could be:

- action
- object
- control

Activities may contain actions of various kinds:

- Occurrences of primitive functions, such as arithmetic functions.
- Invocations of behavior, such as activities.

- Communication actions, such as sending of signals.
- Manipulations of objects, such as reading or writing attributes or associations.

There are actions that invoke activities, either directly using call behavior action or indirectly with call operation action.

### Steps (recipe)

1. Draw a starting point.
2. Draw each step of the process as a rectangle.
3. Add a decision object as a diamond between steps that require a choice to be made.
4. Connect the steps and decisions to display the relationship.
5. Add any additional step or decision objects as needed to complete the model.
6. Draw an ending point.

### Steps to find the Visio

1. Open Visio and select **New**.
2. Enter **UML Activity** or **Flowchart** in the Search box.
3. Open the template to begin creating the diagram.
4. For instructions see, [Create a UML activity diagram](#).

### Example in mermaid

For more information about drawing a flowchart (activity) diagram in Mermaid, see <https://mermaid-js.github.io/mermaid/#/flowchart>.

### Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/activityDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Activity\\_diagram](https://en.wikipedia.org/wiki/Activity_diagram)
- UML - <https://www.uml-diagrams.org/activity-diagrams.html>

## Business Modeling Process Notation diagram (flowchart)

Business Process Model and Notation (BPMN) is a standard way to represent business processes graphically. Visio includes a template that contains the graphical elements described by the BPMN 2.0 specification, following the Analytic conformance class.

### Purpose

Use a BPMN diagram to represent a business process using the BPMN specification.

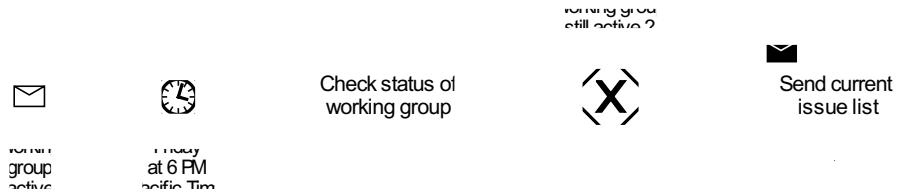
### Type

UML, behavioral

#### NOTE

UML and BPMN are both managed by the same organization, the Object Management Group. Modelers often replace the UML process diagram, the activity diagram, with the BPMN diagram.

### Sample



## Ingredients

- **Start** - The start is the initial condition for the beginning of the flow.
- **Stop** - Stop is the final condition of the flow.
- **Control flow** - The control flow is the line that passes through process steps.
- **Intermediate** - A marker for conditions in the control flow.
- **Process** - A process is a step or activity in the overall flow. A process has an input, an action or set of actions, and an output.
- **Process with sub-processes** - Complex processes can be expanded. In a diagram, a process with sub-processes can be indicated with an icon.
- **Decision** - The control flow may pass through a decision icon.

## Steps (recipe)

1. Draw the start an end condition.
2. Draw each step process until the end condition is reached.
3. When a process a results in different possible outcomes, indicate this using the decision shape.

## Steps to find the Visio

1. Open Visio and select **New**.
2. Enter **BPMN Diagram** in the Search box.
3. For each step in the process you want to model, drag a shape from the stencil to the page, and connect the shapes as usual.

The BPMN specification calls for three types of connectors: Sequence Flow, Message Flow, and Association. Connectors you add by using Auto-Connect or the Connector tool are Sequence Flow connectors by default. You can change the type by right-clicking the connector and then clicking the type you want in the shortcut menu.

4. For general information about connecting shapes, see Add connectors between shapes.
5. Right-click shapes to set the attributes of each object.

Every BPMN shape has some underlying data, or attributes. These attributes specify the appearance of the shape, and additional data associated with the shape. Right-click the shape to view and change the main attributes in the shortcut menu.

For instructions:

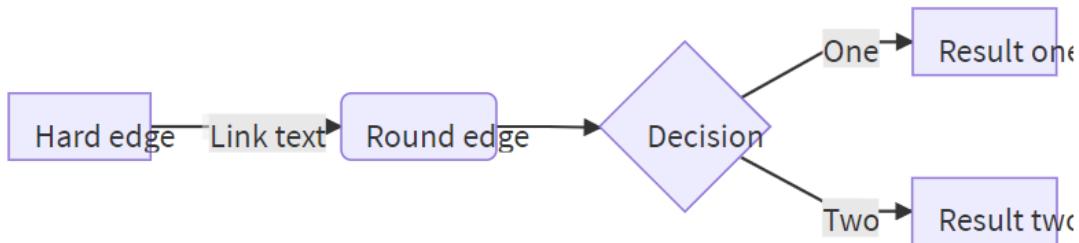
<https://support.microsoft.com/en-us/office/create-bpmn-compliant-processes>

## Example in mermaid

```

graph LR
    A[Hard edge] -->|Link text| B(Round edge)
    B --> C{Decision}
    C -->|One| D[Result one]
    C -->|Two| E[Result two]

```



### Additional information

- Object Management Group and BPMN - <https://www.bpmn.org/>
- Wikipedia - [https://en.wikipedia.org/wiki/Business\\_Process\\_Model\\_and\\_Notation](https://en.wikipedia.org/wiki/Business_Process_Model_and_Notation)

## State machine diagram

A state diagram is a type of diagram used in computer science and related fields to describe the behavior of systems. State diagrams require that the system described is composed of a finite number of states; sometimes, this is indeed the case, while at other times this is a reasonable abstraction.

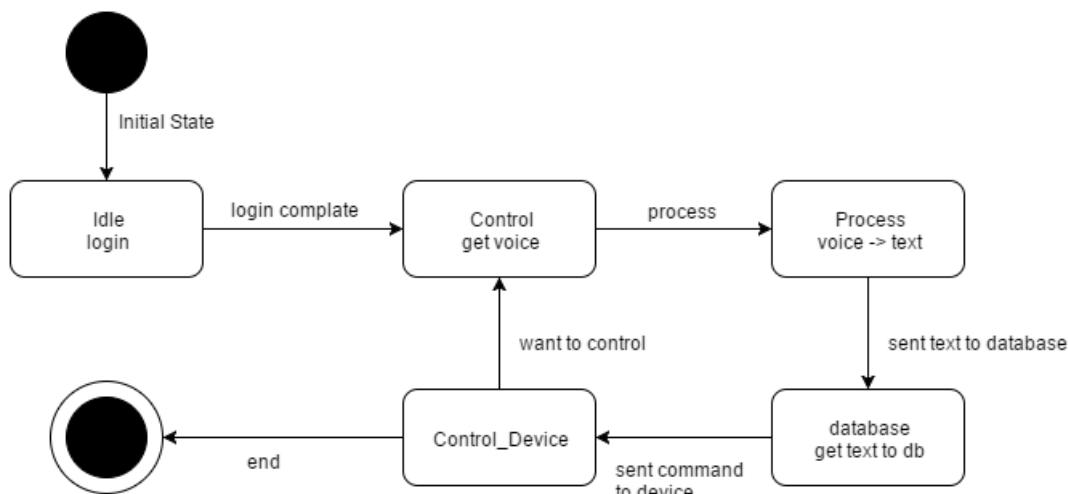
### Purpose

Use a state machine (or statechart) diagram to show the sequence of states an object goes through during its life.

### Type

UML, behavioral

### Sample



### Ingredients

- **Start** - This is a circle.
- **Final State** - This is a circle in a circle.
- **State** - Each state gets a process square. You may label the object and state.
- **Decision** - A decision is a diamond that represents conditional options.
- **Control flow** - The connection from one state to the next is a directed arrow that represents the control

flow.

### Steps (recipe)

1. Select an object that will pass through various states in your system.
2. Add the start icon.
3. Add the object in an initial state. Label the state. As the object changes state, draw a new object and connect it to the previous object.
4. If there is a decision point that may result in different outcomes in state, add a diamond.
5. After the final state object, add the final state icon.

### Steps to find the Visio

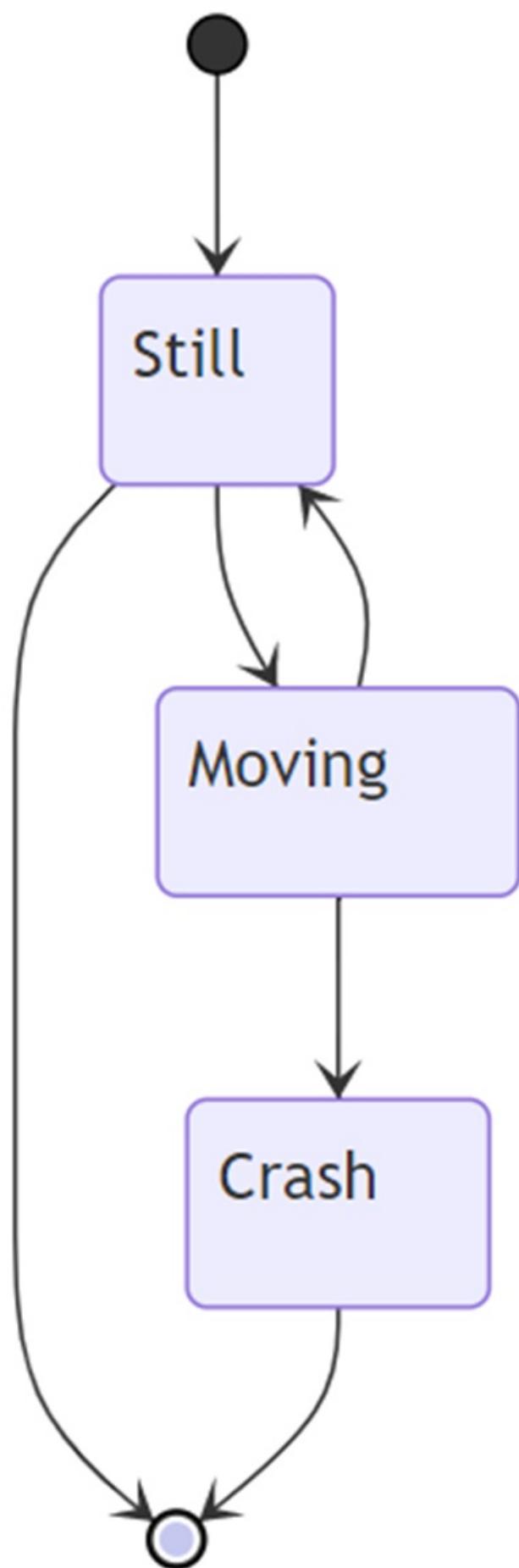
1. Open Visio. Select New and then type UML State Machine.
2. Select the blank template and select Create.
3. The diagram opens. You should see the **Shapes** window next to the diagram. If you don't see it, go to **View > Task Panes** and make sure that **Shapes** is selected. If you still don't see it, click the Expand the Shapes window button on the left.
4. On the **View** tab, make sure the check box next to **Connection Points** is selected. This will make connection points appear when you start connecting shapes.
5. Now, drag shapes you want to include in your diagram from the **Shapes** window to the page. To rename text labels, double-click the labels.

For instructions:

<https://support.microsoft.com/office/create-a-uml-state-machine-diagram-8a681a4d-cf9d-4f57-af07-e91323606366>

### Example in mermaid

```
stateDiagram-v2
[*] --> Still
Still --> [*]
Still --> Moving
Moving --> Still
Moving --> Crash
Crash --> [*]
```



Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/stateMachineDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/State\\_diagram](https://en.wikipedia.org/wiki/State_diagram)
- UML - <https://www.uml-diagrams.org/state-machine-diagrams.html>

## Sequence diagram

A UML sequence diagram shows how a set of objects interacts in a process over time. It shows the messages that pass between participants and objects in the system, and the order in which they occur.

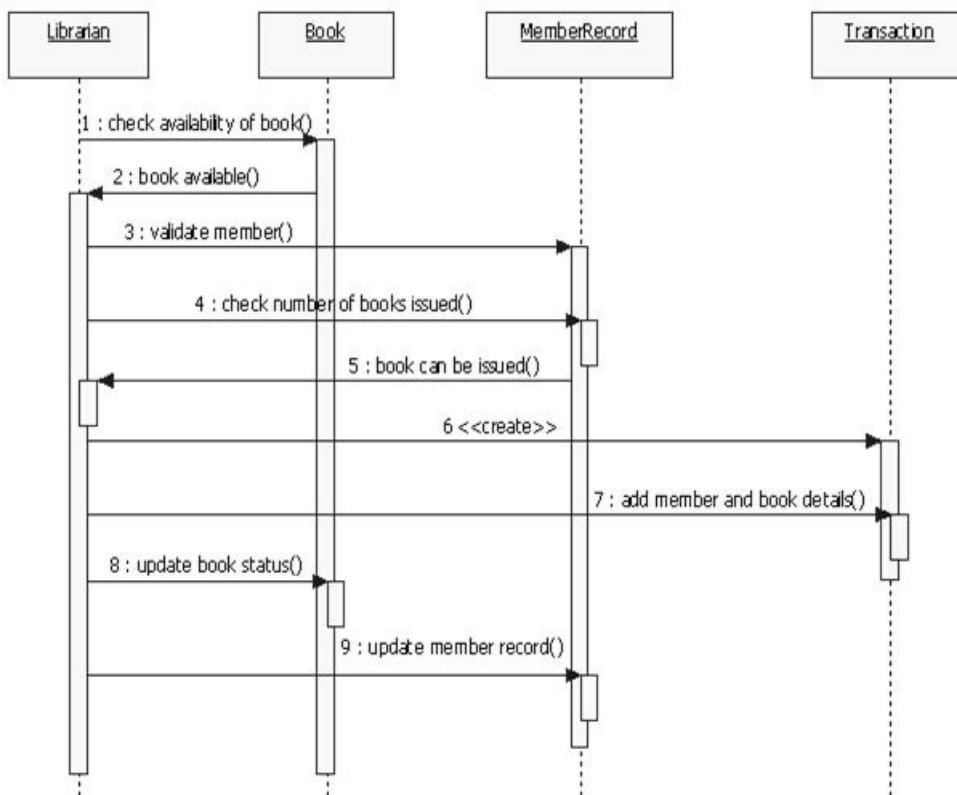
### Purpose

Use the sequence diagram to model the steps in a communication exchange along with object lifecycles and details about the message contents.

### Type

UML, behavioral

### Sample



### Ingredients

- **Object** - This is the thing that is communicating to other things. It might be a user, software object, or machine.
- **Activation** - The vertical line connected to the object and indicates that messages are received or sent by the object.
- **Message** - A message is an arrow on the horizontal axis from one activation line to another. The vertical axis is the timeline. This means a message at above another message is sent before the second.

### Steps (recipe)

1. Identify your objects and place them as blocks across the top of the diagram on a horizontal line.
2. For each object, draw the activation line along the vertical axis.
3. Begin the journey of your message from one object to another by drawing a message arrow.

## Steps to find the Visio template

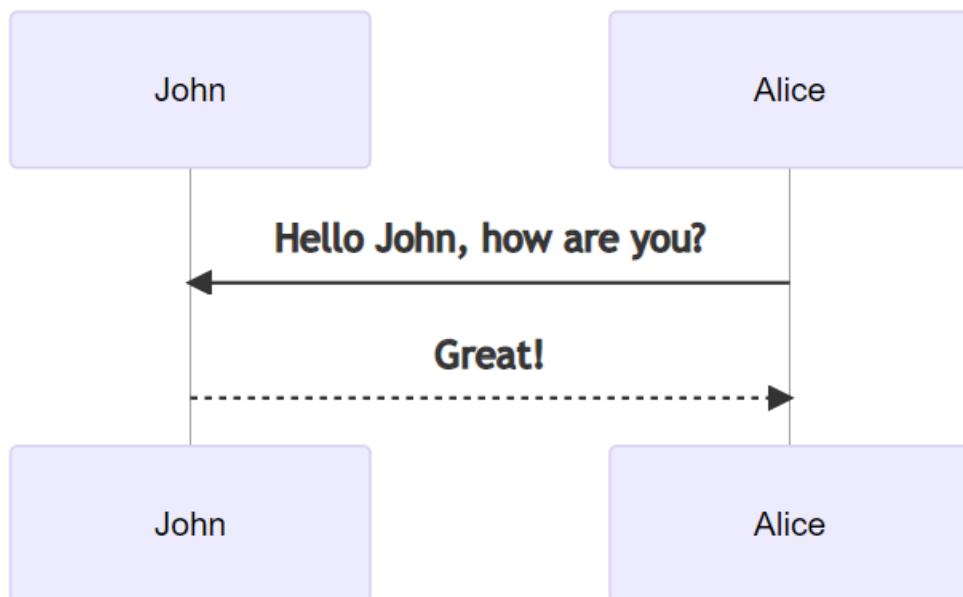
1. Open Visio. Select New and then type **UML Sequence**.
2. Select the blank template and select Create.
3. The diagram opens. You should see the Shapes window next to the diagram. If you don't see it, go to **View > Task Panes** and make sure that **Shapes** is selected. If you still don't see it, click the **Expand the Shapes** window button on the left. 4. On the **View** tab, make sure the check box next to **Connection Points** is selected. This option makes connection points appear when you start connecting shapes.
4. Now, drag shapes you want to include in your diagram from the Shapes window to the page. To rename text labels, double-click the labels.

For instructions:

<https://support.microsoft.com/en-us/office/create-a-uml-sequence-diagram>

## Example in mermaid

```
sequenceDiagram
    participant John
    participant Alice
    Alice->>John: Hello John, how are you?
    John-->>Alice: Great!
```



## Additional information

- Agile Modeling - <http://agilemodeling.com/artifacts/sequenceDiagram.htm>
- Wikipedia - [https://en.wikipedia.org/wiki/Sequence\\_diagram](https://en.wikipedia.org/wiki/Sequence_diagram)
- UML - <https://www.uml-diagrams.org/sequence-diagrams.html>

## Next steps

Review [Diagrams for technical communicators](#).

# Process mapping overview

4/7/2021 • 2 minutes to read

A process map is a picture of steps, decisions, and other elements of either a work or mechanical operation. This article provides tools for defining and summarizing a process.

## How to map a process

A process is any set of repeated actions that take something and produce something else. The material that is transformed by the process is called input. And the product of the process is called the output. Between the input and output, there are many steps, decisions, and one or more actors.

A *process map* represents the work used to make something. It can be helpful to mindful that the process itself is not the work. The work changes, and the process may need to change to reflect these updates.

1. Identify the set of actions as a process. Use the process definition form to find the process boundaries, who are involved in the process, the process dependencies, and the expected product of the process.
2. With a defined process, you may need to create a formal or informal process definition workgroup. The workgroup is made of the people who perform the process, stakeholders in the process, people providing the input to the process, and people receiving the output of the process. A formal group may define itself as working on mapping the process. An informal group may be coordinated by the writer as they have questions about different aspects of the process.
3. Map the process with feedback from members of the process group.
4. Validate the final draft of the process map.
5. Once defined, a process may need to be maintained and changed as the context of the process changes.

## Tools for defining a process

This section provides the following items to help you define a process:

- **Process definition form**

A process definition form help you focus on the boundaries of your process and collect the information you need to create a process summary or process map. This is a good place to start when you are researching your process or preparing to create a process definition workgroup.

- **Process summary**

A process summary is an analysis and map of the overall process. The process summary provides detail in each step of the process. The process summary can be the product of a process definition workgroup.

- **Process map**

A process map contains the diagram representing the process and essential information to support the process. The process map is meant for quickly documenting a work process or assessing a process for automation.

## Diagrams for process models

The Diagram cookbook contains four diagram types that are often used to represent processes.

- **Data flow diagram**

A data-flow diagram is a way of representing a flow of data through a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the

process itself.

- **Activity diagram**

An activity diagram shows objects and choices representing the workflow of stepwise activities. This type of diagram is used to show an activity process from start to finish.

- **Business Modeling Process Notation diagram**

Use a BPMN diagram to represent a business process using the BPMN specification.

- **State machine diagram**

Use a state machine (or statechart) diagram to show the sequence of states an object goes through during its life.

## Next steps

Diagrams for technical communicators.

# Process definition form

4/7/2021 • 2 minutes to read

A process definition from help you focus on the boundaries of your process and collect the information you need to create a process summary or process map. A process summary is an analysis and map of the overall process. A process map contains the diagram representing the process and essential information to support the process.

You can use the following information to define your process.

## Process name

**Process type:** [as is, reengineer, new] (Choose one.)

- **As is** is an effort to document a previously undocumented process.
- **Reengineer** is to review an existing process for process improvement or to capture changes in the process.
- **New** is an effort to create a process where one does not currently exist.

**Date:**

**Problem/goal:** What problem or goal do you hope to address by analyzing the process?

**Process output:** The product or services that are created by the process; that which is handed off to the customer.

**Output users:** The person or persons who USE your output—the next in line. Whether your customers are internal, they use your output as an input to their work process(es).

**User requirements:** What the end user of the process needs, wants and expects of the output. User's generally express requirements around the characteristics of timeliness, quantity, fitness, ease of use, and perceptions of value.

**Process players:** These are the people who actually do the steps of the process—as opposed to someone who is responsible for the process, such as the process owner/manager.

**Process owner:** The person who is responsible for the process and its output.

**Stakeholders:** A process stakeholder is someone who is not a supplier, customer, or process owner, but who has an interest in the process, and stands to gain or lose based on the results of the process.

**Process margins:** The first step (receiver of input) and last step (producer of output) of the process.

**Input:** The materials, equipment, people, money, or environmental conditions that are required to carry out the process.

**Supplier:** The people (functions or organizations) who supply the process with its inputs.

**Process performance:** How is the performance of the process measured (metric)?

## Next steps

[Process mapping overview.](#)

# Process summary format

4/7/2021 • 2 minutes to read

The following document is a framework for assessing and documenting your process effort and the process. You can use the [process definition form](#) to prepare information for this process.

## Process summary

Summary of the process. Write the definition of the process by answering the problem or goal do you hope to address by analyzing the process.

## Process name

Type the name of process owner.

## Summary

This process <what it does>. Input. Process. Output.

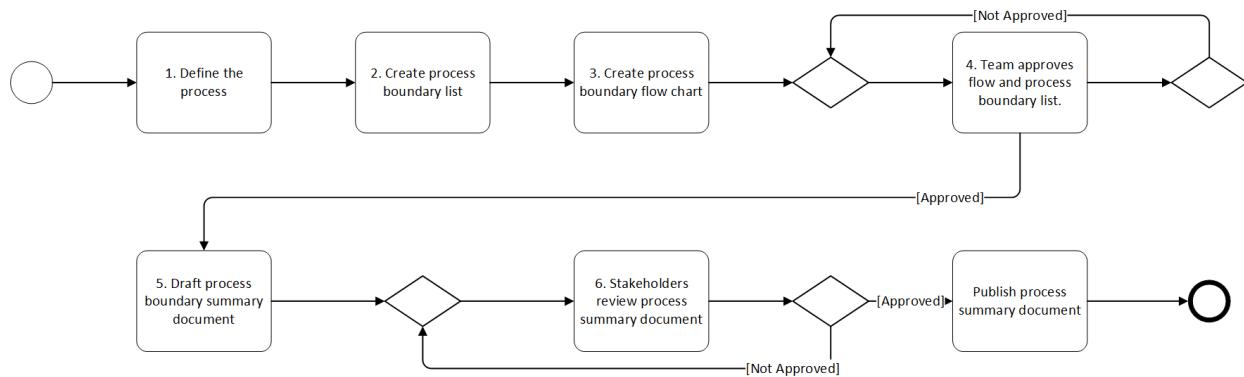
## Overview

For each step in your process, update the following lines:

- With the **input** the **role step** to **output**.
- With the **input** the **role step** to **output**.
- With the **input** the **role step** to **output**.

## Process boundary flow

In this position, add a process map to your process summary.



## Process boundary list

For each step in your process, update the following rows of the table. You will break down what the step is, who does the step, what is handed off in the step and what is produced in the step.

For example:

#	STEP	ROLE	INPUT	OUTPUT
1	Print a hard copy of the document	Writer	Document File (Writer)	Paper document

Complete the following table with the number of steps in your process:

#	STEP	ROLE	INPUT	OUTPUT
1	Verb noun	Role (job title)	Noun (role that provided the hand off)	Noun Noun
2	Verb noun	Role (job title)	Noun (role that provided the hand off)	Noun Noun
3	Verb noun	Role (job title)	Noun (role that provided the hand off)	Noun Or Noun

## Process details

These lines can be defined with the process definition worksheet. Define the material, equipment, information, people, money, or environmental conditions required to perform the process.

ITEM	DEFINITION
<b>Input:</b>	The materials, equipment, people, money, or environmental conditions that are required to carry out the process.
<b>Output:</b>	The product or service that is created by the process.
<b>Problem/goal:</b>	What problem or goal do you hope to address by analyzing the process?
<b>Requirements:</b>	What the end user of the process needs, wants and expects of the output.
<b>Process Owner:</b>	Name
<b>Stakeholders:</b>	A process stakeholder is someone who is not a supplier, customer, or process owner, but who has an interest in the process, and stands to gain or lose based on the results of the process.
<b>Process margins:</b>	Role and Step at the beginning and end of the process. Where does it start and where does it end.
<b>Metric:</b>	How is the performance of the process measured (metric)?

## Process team members

NAME	ROLE
Name	Title
Name	Title
Name	Title

## Next steps

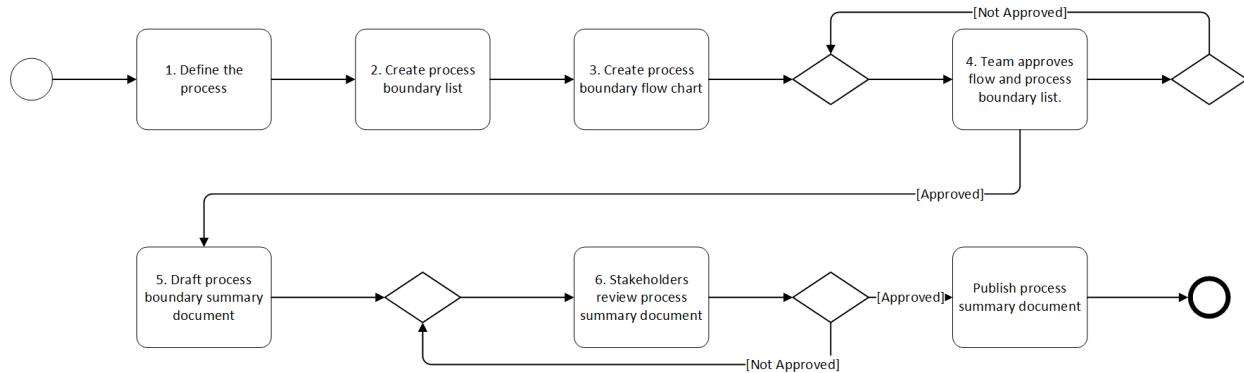
[Process mapping overview.](#)

# Process map

4/7/2021 • 2 minutes to read

Shows the actions of the agents and actors in the application.

## Diagram



## Activity Description

ACTIVITY NAME	NUMBER ON THE GRAPH.
Related to Use Case	Typically verb and direct object.
Created By	<yourname>
Date Created	<date>
Priority	High, Medium, Low
Actor	Name of the Actor
Secondary Actor(s)	Name of Secondary Actor(s).
Description	State the context of the use case and the goal of the use case.
Pre-conditions	State the required inputs into the process captures by the use case.
Successful End Condition	State the required outputs and who may be expecting the output from the use case.
Failed end Condition	What are the expected events when the process fails.
Frequency of Use	Continual, Hourly, Interval, Daily, Weekly, Monthly, Annual
Main Flow	Step 1 Step 2 Step 3

ACTIVITY NAME	NUMBER ON THE GRAPH.
Alternative Flow	Step 1 Step 2 Step 3

## Next steps

[Process mapping overview.](#)

# Diagrams as code

4/7/2021 • 5 minutes to read

The processes in this guide support drawing diagrams in any tool that works for you such as a Sharpie on a whiteboard, pen and paper, Visio, or even PowerPoint. An alternative diagramming workflow takes the separation of the diagram from presentation a step further. With *diagrams as code*, you can describe your diagram in a text format such as Graphviz's DOT language, an Excel file, or markdown syntax designed for describing graphics.

For diagrams as code, a drawing is broken into text-based data that can be stored in GitHub and managed with versions and commits in the same way as any other Content and Learning document. This code serves as the basis for diagrams and views that generate diagrams.

## Elements of diagrams as code

You need the following items to create a workflow using code for your drawings:

**A document format.** To create a doc as code workflow, you need your graph to be stored as a document that can be both read and written by a person but also parsed by an application. Common formats include markdown, XML, yaml, and JSON. The source document can be contained in a version control system such as Git.

**A rendering engine.** An application can then transform the document into a graph. In a system that uses static site builders, you can run this transformation using a command-line tool or dedicated tool such as yEd or Gelphi. Point at the source-controlled document describing the graph and then render the graph in a display format.

**The display format.** The product site uses portable network graph (PNG) or scalable vector graphic (SVG). You may generate multiple diagrams from a source code document.

## Example workflow for diagrams-as-code

A workflow for using diagrams-as-code for [docs.microsoft.com](https://docs.microsoft.com):

1. Create your document in a format friendly to your rendering engine and to your own authoring practice.  
Store the document in a GitHub repository.
2. Run the command-line tool that can trigger your rendering engine. Point to the source document and save the target document into your media folder.
3. Add the media reference to your markdown file.

When you have changes to your diagram, you can update the code that describes your graph, rebuild the diagram, and update the media file.

The following four are examples of a diagram as code tool chain.

NAME	DOCUMENT FORMAT	RENDERING ENGINE	OUTPUT FORMAT
Graph tool	Comma-separated-files (CSV)	yEd or Gelphi	SVG, PNG
PlantUML	PlantUML syntax	PlantUML Server	SVG, PNG

Name	Document Format	Rendering Engine	Output Format
Mermaid	Mermaid syntax	Mermaid.js	SVG, PNG
Graphviz	Dot	Graphviz	SVG, PNG

## Use a graph tool

In this workflow, you will create two tables describing your diagram. One table will be the list of things. The other table will be the relationships between the things. You can use then yEd Graph Editor from yWorks as your rendering engine.

If you do not have yED installed, install yEd. <https://www.yworks.com/products/yed>

### Create and export a file

1. Create a list of the things you are going to diagram. Each thing must have a unique ID and a name. Save as a CSV named, diagram-things.csv.
2. Create a list of pairs of things with a relationship. Each relationship must connect two things. These things are shown by their IDs from the thing list. Save a CSV named diagram-relations.csv. These two documents can be in source control.
3. Combine your diagram into an Excel workbook with each CSV as a table in its own worksheet. Save your file in a temporary location.
4. Open the yEd Graph Editor, select **Open** and then select the workbook.
5. In the MS Excel Import window, select your Things as the Nodes. a. Select the **Data** range from your things table. Select **Adopt**. b. Select the column that contains your things unique ID. Select **Adopt**. c. If you have defined a group column with unique names for each group, select the group column and then select **Adopt**.
6. Select your relationships as your Edges. a. Select the Data range from your relations table. Select **Adopt**. b. Select the column that contains your relationships. Each relationship has a source and a target that specified a node ID. Select **Adopt**.
7. Select the presentation tab in the MS Excel Import window.
8. Select the column with the name of the node.
9. Select **Layout Organic**.
10. Prepare your graphic using yED's presentation options and then save it as an SVG to your repository.

### Learn more about graph tools

- You can find more information about yEd at the [yWorks website](#).
- [Gelphi](#) is a Graphviz tool with a good set of features for analyzing network graphics.

## Use PlantUML

In this workflow, you will create a text file using the PlantUML drawing language. You can store this text file in your version repository. You can then use a PlantUML rendering engine to draw the diagram and place the output format into your documentation.

In this workflow example, you will use the PlantUML web server at the following location:

<http://plantuml.com/guide>

### Create and export a file

1. Create your diagram as text. For example, to create a sequence diagram:

```
@startuml
actor Foo1
boundary Foo2
control Foo3
entity Foo4
database Foo5
collections Foo6
Foo1 -> Foo2 : To boundary
Foo1 -> Foo3 : To control
Foo1 -> Foo4 : To entity
Foo1 -> Foo5 : To database
Foo1 -> Foo6 : To collections
```

2. Open the PlantUML server: <http://www.plantuml.com/plantuml/> and cut and paste your text into the server.

3. Place the output format into your documentation.

### Learn more about graph tools

You can learn about writing PlantUML docs in [PlantUML in a nutshell](#).

## Use Mermaid

Mermaid is a Javascript-based diagram tool that uses the code as diagram mode and was inspired by markdown. It works well in a markdown workflow and a rich ecosystem of Visual Studio Code and GitHub support.

Before you follow this workflow, you will want to install the mermaid plug-ins to Visual Studio Code. Install the following extension:

- **Mermaid Preview Support**

<https://marketplace.visualstudio.com/items?itemName=bierner.markdown-mermaid>

- **Mermaid Export**

<https://marketplace.visualstudio.com/items?itemName=Gruntfuggly.mermaid-export>

### Create and export a file

1. Create a file in Visual Studio Code named `diagram-flow.mmd`.

2. Copy and paste the following Mermaid syntax into the file:

```
graph TB
    A --> C
    A --> D
    B --> C
    B --> D
```

3. Select the file and open the mermaid preview.

4. You can export the file as a `diagram-flow.svg` and place the output format into your documentation.

### Learn more about graph tools

Learn about how Mermaid lets you create diagrams using [text and code](#).

## Use Graphviz

In this workflow, you will create a text file using the DOT language. The DOT file is a text file that describes the elements of a diagram and their relationship so that the tool can produce a graphical representation. You can

then Graphviz as your rendering engine to draw the diagram for you and place the diagram into your repository.

If you have not installed Graphviz and made it available through your binary, you can download it from the following location: <https://www.Graphviz.org>

### Create and export a file

1. Create a graph file named diagram-thing.dot. You can create a server topology with the following example:

```
graph example1 {  
    Server1 -- Server2  
    Server2 -- Server3  
    Server3 -- Server1  
}
```

2. From the command line, run the following command:

```
dot diagram-thing.dot -Tpng -o diagram-thing.png
```

### Learn more about graph tools

Learn more about the open-source [Graphviz software](#).

## Next steps

Review [Diagrams for technical communicators](#).

# About technical diagram and art

4/7/2021 • 4 minutes to read

This set of articles focuses on technical diagrams. Art is the larger category of visuals that might be found in a document.

We'll begin with a brief definition: a technical diagram shows the composition and relationship of things. There are two types of relationships. Relationships are connections between things. And things can belong to a group or subgroup.

The primitives of a diagram are:

- Things
- Relationships between things
- Groups of things and groups

The basic element of a diagram is the thing. A thing can be an object such as a machine or a software object, such as an application or a class. Things are the same as nouns. They are a person, place, or object.

The second primitive is a relationship. Relationships depend on things. Two things can be related. For instance, one thing might send data to another thing. In a diagram, you illustrate two things as boxes with a relationship as a line between the two boxes.

The third primitive element is a grouping. Two things might belong to a system and you would draw a box around the two things that belong to the same system.

With the elements of thing, relationship, and group you can build maps and blueprints of complex system architectures, intricate communication patterns, and class diagrams showing the object model of a software application.

The difficulty comes in assessing the drawing and what the drawing depicts. You must keep the thousand different ways you can draw a box, a line, and a group consistent from diagram-to-diagram. And, another difficulty comes in keeping in mind the limits of a reader. Often the reader encounters the diagram when learning about a subject that is new to them. A reader can only perceive a limited set of things in a single view. They can quickly become overwhelmed by a poster-sized diagram with hundreds of things, relationships, and groups.

The procedures in this collection of articles introduce a process for getting started with a subject and for refining a sketch of the subject into a formal diagram. You can also use these steps to assess and revise diagrams that have been provided to you by your technical partners.

This procedure distinguishes a diagram from a chart or data visualization. A diagram such as a system architecture diagram differs from a scatter chart, for instance. Both things are broadly art, but they have different purposes and are composed of distinct elements. In *The Art of Business Processing Modeling*, Martin Schedlbauer notes:

Modeling imposes structure and isn't simply a drawing. Proper modeling imposes a certain intellectual rigor on the artifacts being created and visually represented. This is one of the reasons that the use of a standard visual modeling language is critical.

This set of articles uses the Unified Modeling Language, Business Process Modeling, and extends these models with a few common diagram types such as data flow diagrams, network diagrams, and entity relationship

diagrams.

While you can get started now, you may want to consider some operating principles behind these articles.

## Separation of concerns

This process works toward separating the information or data that you are diagramming from its presentation.

A diagram often depicts complex systems with a lot of detail. The technical diagram is concerned with technical thoroughness and accuracy.

Yet this process is at odds with creating a presentation that can provide a quick conceptual understanding for a reader about a complex system. This is where an illustration can be useful. A technically accurate illustration is often based on a complex diagram. The illustration compresses and leaves out information and focus on providing the minimum information needed to communicate a concept. By separating these concerns, the diagram can focus on the data and information when you're researching and learning about your subject. And then you can focus on visual communication when creating the diagram used in the documentation.

## Agility

This process isn't specific to Agile or Waterfall methodologies. This process has been adapted to model-driven design, an Agile method, but also references and uses standards and diagram types that have their roots in some classic technical design disciplines such as object-oriented design and systems analysis. These approaches were used in what we now call Waterfall. You can create a rough diagram in 10 minutes using this process, or you can spend hours creating a dense model with multiple views of a complex system.

## One Microsoft

As a formal approach, this process supports a standardized diagramming language. We spell out the basic elements and syntax of diagrams. And then we talk about the formal diagrams that use these elements for specific purposes. And finally, these diagrams can be presented as an illustration using the Microsoft design standards and library of icons. The intended result is to create diagrams that speak with the authority of Microsoft technical documentation and can be easily understood by readers fluent in the most common diagramming standards. Our readers will not have to spend their time trying to sort out how a diagram explains a topic every time they encounter one. Instead, they can focus on the meaning of the diagram.

## Open tools

By focusing on conventions that you can use with pen and ink, these procedures anticipate that you may have your own tools that you would like to use. If you can draw the diagram on a whiteboard, then you can easily use Visio, PowerPoint, or a diagram-as-code solution. If you have the drawing skills to draw a shape that has four sides and a line connecting two points, then you have the drawing skills to draw a good diagram.

## Next steps

Review [Diagrams for technical communicators](#).

# Use style guides for content on docs.microsoft.com

4/12/2021 • 2 minutes to read

When you're writing or editing Docs content, you can turn to several internal resources for standards on formatting, terminology, and style. If you've ever scoured these resources only to find conflicting guidance, you might have wondered which guide you should follow. To determine which resource takes precedence, use the hierarchy described in this article.

## Hierarchy of guides

Our [federated](#) hierarchy concept can help you decide which style guidance to follow. This list starts at the most local level and moves to the most general (federal) level.

Start at the top of this list and move down to find the most authoritative guidelines for your context:

1. Your product or service style sheet (if one exists). For more information, see the [Guidance for your service or product](#) section in this article.

### NOTE

Service-level style guidance for the Content & Learning (C&L) organization is located in the "Repo-specific guidance" section of the Docs contributor guide.

2. The [contributor guides](#) are your starting point because they're tailored to your organization's content. Check to see whether the guides address your question before you move on to other resources.
3. [Microsoft Cloud Style Guide](#) is the official resource for cloud-specific names, standards, and terms.

If you can't find a product, service, or feature name here, search for overview and pricing pages on Docs. The [directory of Azure products](#) is good place to start your search. Related articles on Docs can give you an idea of names that are commonly used, but these articles aren't reliable authorities.

4. [Microsoft Writing Style Guide](#) is the overarching style guide for Microsoft. It provides details on standards (voice, word choice, and formatting) across Microsoft documentation. It's not specific to Docs. But the Writing Style Guide is the basis for the Cloud Style Guide, so the two guides dovetail well.
5. All other Microsoft style guides. When you don't find a standard in the Docs contributor guide, Cloud Style Guide, or Writing Style Guide, try searching the [main site for Microsoft style guides](#). You might find an answer that's helpful even though it's not a standard for Docs content.

## Guidance for your service or product

When you're writing or editing Docs content, you might need standards beyond what the guides provide. To fill in the gaps left by the guides, you can create a style sheet to track your decisions about the conventions and terminology specific to your product or service. Your style sheet is the most [local level in the federated hierarchy model](#).

A style sheet fills in gaps in existing resources as conventions and terminology evolve. Just keep in mind that you shouldn't waste resources by documenting standards that the guides already cover. Also avoid creating style guidance that conflicts with rules at a higher level in the hierarchy, unless you have a good reason to diverge from the guidance.

# Starting points

To learn about the resources each guide provides, explore some of these helpful starting points:

- Docs contributor guide:
  - [Text formatting guidelines](#)
  - [Repo-specific guidance, starting with Azure](#)
  - [Everyday word list](#)
  - [Terminology and inclusive language](#)
- Cloud Style Guide:
  - [A–Z names + terms dictionary](#)
  - [Grammar + usage](#)
  - [Legal \(CELA\) considerations](#)
- Writing Style Guide:
  - [Top 10 tips for Microsoft style and voice](#)
  - [Acronyms](#)
  - [Capitalization](#)
  - [Writing about procedures and instructions](#)

# Next steps

If our internal guides are silent on the question you're researching, check the standards in these references:

- [American Heritage Dictionary](#)
- [Chicago Manual of Style](#)

# Technical principles checklist

4/16/2021 • 2 minutes to read

This checklist is designed to help you evaluate your article's content per the [Technical Focused Review](#) (TFR) principles.

When considering items in the checklist, take into account the [content type](#) being evaluated. Not all items in the checklist are applicable to all article types. For example, "Provide technological choices" might not be appropriate for a quickstart.

## Functionality

- Technical steps are correct and functional.
- After following the steps, does the customer understand what they did, and why?
- Are all prerequisites explicitly defined?
- Are resources allocated in a cost-effective manner? Are they cleaned up (if applicable)?

## Technologies

- Are the most appropriate services, technology, and tools used for the task?
- Are the latest features of the service or product used?
- Does the article provide technological choices? For example, if there is an alternative method or technology that's applicable, is that presented?
- Do the technical steps present best-practice methods and production-ready examples for completing the task?
- If best-practice methods and production-ready examples **can't** be presented (due to the article's content type, for example), does the article clearly explain the implications of the "bad-practice" example? Does it provide pointers to the recommended best-practice way of doing things (if appropriate to content type)?

## Code and scripts

- Has your code been reviewed, such as with CodeFlow or other review process?
- Is your code using the most current features of the language?
- Is your code idiomatic? For example, are you writing Python like Python and not C#?
- Do code and script samples follow established [style and conventions](#)?

## Learning goals

- Does the customer leave the article understanding its context within the larger ecosystem?

## Code conventions

- [Azure CLI](#)
- [Azure PowerShell](#)
- [C# Coding Conventions \(C# Programming Guide\)](#)

# Azure CLI and Azure PowerShell best practices

4/16/2021 • 3 minutes to read

This document describes some general best practices for writing scripts for user consumption as examples and documentation. It's **non-exhaustive** but provides guidance to help you get started and handle common situations. Most suggestions in this article come from following the idea behind [the principle of least surprise](#).

## List required Azure resources, preview modules, and extensions

Make sure requirements like pre-created resources, CLI extensions, or preview versions of an Azure PowerShell module are called out near the top of the script. CLI extensions and preview modules must also include their version.

### CLI example

For a script that requires you to have an existing Linux VM and to have the `virtual-wan` extension version `0.1.0` installed:

```
# This script requires you have the following resources created:  
# - An Azure Virtual Machine running Linux  
  
# This script requires the following Azure CLI extensions:  
# - virtual-wan (0.1.0)
```

### Azure PowerShell example

For a script that requires you to have an existing Blob Storage container and the `Az.Storage 1.3.2-preview` module:

```
# This script requires you have the following resources created:  
# - An Azure Blob Storage container  
  
# This script requires the following preview module:  
# - Az.Storage (1.3.2-preview)
```

## Make requirements clear

If your script requires a specific OS, software version, installed/enabled feature, etc. then make the requirement clear in a comment. Keep in mind that PowerShell 6.x and later are cross-platform products standardized on .NET Standard. Using .NET Foundation features requires Windows.

This includes requirements such as permission levels on Azure. For example, if your script requires that the user have `admin`-level privileges on the tenant, make sure this is made clear.

For CLI scripts, try to avoid as many non-`az` commands as possible. Stick to basic UNIX system utilities unless there's a good reason not to, such as [using openssl for password generation](#).

## Keep run times reasonable

There are a large number of scripts which have runtimes that users may consider "unreasonable" for a sample. Please keep in mind that the scripts provided are often used as cookbooks or other modified starting points. If your script takes an extremely long time to run, ask yourself:

- Should this even be a sample? Is the process complicated enough that users need this guidance as a sample and not a how-to or other conceptual documentation?
- Which commands contribute the most to the length of execution? Can they be removed, replaced, or made faster? Should users be given an estimate of execution time?

## Explain expected errors

When a user sees an error, a common assumption is *this script doesn't work*. If you expect your script to error out under certain acceptable circumstances, make sure that the error is clearly explained comments. Documenting these errors helps ease user frustration.

For example, CosmosDB APIs return "404 Not Found" for undefined properties. That error is easy to misread as "database doesn't exist." A comment before the command explains the error:

```
# Get the throughput for a keyspace (returns RU/s or 404 "Not found" error if not set)
Get-AzResource -ResourceType $keyspaceThroughputResourceType `
    -ApiVersion $apiVersion -ResourceGroupName $resourceGroupName `
    -Name $keyspaceThroughputResourceName | Select-Object Properties
```

## Don't unnecessarily modify the user environment

Don't set variables that your script doesn't use! Keep in mind that users getting scripts from the documentation site will be using cut and paste. Don't use the `export` keyword when setting variables in `bash` or set properties on PowerShell's `$env`: It may modify the user's environment in ways they aren't expecting.

## Don't install software

One of the most invasive ways to modify a user environment is by installing software without their consent. Don't automatically install CLI extensions, preview modules, or other external software. Instead provide a prompt to install or tell users to install manually in a comment.

# Azure CLI article guidelines

5/5/2021 • 9 minutes to read

This article describes the style convention for writing Azure CLI articles. Following this guidance ensures that content for the Azure CLI is consistent. For more information on submitting standalone scripts and the pull request process, see:

- [Azure CLI and Azure PowerShell best practices](#)
- [Azure tools script PR review process](#)

The [Azure CLI editor's checklist](#) provides a list of *minimum requirements* for new and revised Azure CLI articles.

## Table of contents

- **name**: Omit extra words like **Azure CLI** and **Quickstart** if this information is clearly identified in the [TOC file format](#). Instead, take the opportunity to provide more information about the function of the article.
- **displayName**
  - Add this property whenever possible adding CLI reference commands. Example: `az dt` and `az maps` are used by Azure IoT but may not be found in a TOC search.
  - There isn't room for most TOC entries to say "[article name] for portal and Azure CLI". Use the **displayName** property to identify every article in a TOC that contains CLI syntax.
  - Use logical judgment as most CLI articles contain **az login** and **az group**. Only add reference commands that are unique to the article intent and Azure service.
  - Example value: `displayName: Azure CLI, az iot, az dt, az group, az account`
- **href**: Follow [hyperlink guidelines](#).
  - The Azure CLI reference and cross-service doc content are stored in a dedicated GitHub repository.
  - Adding `?view=azure-cli-latest` to Azure CLI links is not necessary.
  - Example links:  
**Good:** `/cli/azure/install-azure-cli`  
**Avoid:** `https://docs.microsoft.com/cli/azure/install-azure-cli`  
**Incorrect:** `/cli/azure/install-azure-cli ?view=azure-cli-latest`  
**Incorrect:** `https://docs.microsoft.com/en-us /cli/azure/install-azure-cli ?view=azure-cli-latest`
- **Reference node**: Add a link in the Reference node of the TOC to the Azure CLI reference content.

## Article metadata

- **ms.custom:** must contain the tag **devx-track-azurecli**
- **ms.date:** must be in the format **MM/DD/YYYY**.
  - Change the date when there is a significant or factual update.
    - Reorganizing the article
    - Fixing factual errors

- Adding new information
- Do not change the date if the update is insignificant.
- Fixing typos and formatting
- `title:` must be a unique string of 43-59 chars including spaces.
  - Do not include site identifier (it's auto-generated).
  - Use sentence case capitalizing only the first word and any proper nouns.
- `description:` should be 115-145 characters including spaces.

## Prerequisites

### Include files

- The [Use Azure Cloud Shell](#) include file is no longer recommended for Azure CLI docs. Instead, use one of three Azure CLI prerequisite include files when using three or more CLI references in an article.
  - The [CLI "Prerequisite" with H2 header](#) is best used when code blocks contain only Azure CLI syntax.
  - The [CLI "Prepare your environment" with H3 header](#) is designed to be used in conceptual tabs.
  - The CLI [include file with no header](#) works well when CLI follows other prerequisites in a bulleted list.
- Always place Azure service prerequisites first *followed by* instructions for Azure Cloud Shell and the Azure CLI. For example, "You must have `Microsoft.Authorization/roleAssignments/write` permissions to complete the instructions in this tutorial" should come before "Prepare your environment for the Azure CLI".

## Reference content

### Versions

- If the reference command requires a minimum version of the CLI higher than 2.0, mention the version in your article and sample script.
- All extension references are version-specific and often require the most recent [Azure CLI release](#).
- Double-check core references although most only require version 2.0.67 or later.

### Links

- Supply a link to the Azure CLI reference content as often as possible without being redundant.
- There should be at least one reference link for every CLI reference group within an article.
- Avoid linking to a large A-Z reference list that forces the user to scroll to find a reference command.

Here are a few examples:

	EXAMPLE	USE WHEN
Reference group:	<a href="#">az eventhubs</a>	You're writing about a command group and not referring to a single reference or command.
Reference:	<a href="#">az eventhubs namespace</a>	You're writing about a reference and not referring to a single command.
Reference command:	<a href="#">az eventhubs namespace create</a>	You're writing about a reference action.

### Content

## IMPORTANT

Do not copy reference content from the auto-generated docs found in <https://docs.microsoft.com/cli/azure/> or in <https://docs.microsoft.com/cli/azure/ext/>. Instead, provide a link to the reference article.

## Good:

"Azure CLI commands used in this article (H2)"

- [az group create](#)
- [az network vnet create](#)
- [az network vnet subnet create](#)
- [az network vnet subnet show](#)
- [az cosmosdb create](#)
- [az group delete](#)

**Incorrect:** (Notes have been copied from reference content, and although not likely to change, these descriptions will not be auto-updated.)

"Azure CLI commands used in this article (H2)"

COMMAND	NOTES
<a href="#">az group create</a>	Creates a resource group in which all resources are stored.
<a href="#">az network vnet create</a>	Creates an Azure virtual network.
<a href="#">az network vnet subnet create</a>	Creates a subnet for an Azure virtual network.
<a href="#">az network vnet subnet show</a>	Returns a subnet for an Azure virtual network.
<a href="#">az cosmosdb create</a>	Creates an Azure Cosmos DB account.
<a href="#">az group delete</a>	Deletes a resource group including all nested resources.

## Syntax formatting

Use the following short-list of applied guidelines that are designed to coincide with [PowerShell syntax elements](#):

1. Always use the full name for reference commands and parameters. Avoid using aliases unless you're specifically demonstrating the alias.
  - **Correct:** `az storage account create --name mySG --resource-group myRG`
  - **Incorrect:** `az storage account create -n mySG -g myRG`
2. Use bold for **command group names**.
  - **Correct:** This reference is part of the **baremetal-infrastructure** extension for Azure CLI and requires version 2.12.0 or higher.
3. Use backticks for `reference names` both partial and full. When writing a conceptual article (as opposed to reference content), the first instance of a reference command name should be hyperlinked to the reference documentation. Don't use backticks, bold, or other markup inside the brackets of a hyperlink.
  - **Good:** Use `az group create` to create a resource group.
  - **Better:** Use [az group create](#) to create a resource group.

- **Incorrect:** Use `az group create` to create a resource group.
4. Use backticks when showing the use of a `--parameter` within text.
- **Correct:** Use the `az group create` command specifying your desired resource group name in the `--name` parameter.
  - **Incorrect:** Use the `az group create` command specifying your desired resource group name in the `name` parameter.
  - **Incorrect:** Use the `az group create` command specifying your desired resource group name in the `name` parameter.
5. Use backticks for parameter `values`.
- **Correct:** Use `az group create --name MyResourceGroup` to create a resource group named `myResourceGroup`.
  - **Correct:** Create a resource group with `az group create`. This resource group is named `myResourceGroup` and the location is `eastus2`.
  - **Incorrect:** Use `az group create --name MyResourceGroup` to create a resource group named `myResourceGroup`.
6. Use bold when talking about a parameter by **name** referring to it as an **object**.
- **Correct:** The `az group create` command has several optional parameters including **managed-by** and **subscription**.
7. Use backticks for `file paths`, and `inline syntax examples`.
8. Use backticks for `URLs` that are not meant to be clickable in the document.
9. *Use italics for emphasis*, not for semantic markup.

## Argument format

Use the long format (`--resource-group`) instead of short format (`-g`) for all arguments.

- **Good:** `az group create --location westus --group MyResourceGroup`
- **Avoid:** `az group create -l westus -g MyResourceGroup`

## Argument order

Arguments to CLI commands should appear in the following order:

1. Resource group name (`--resource-group`)
2. Resource name or identifier (`--name`, `--id`, etc.)
3. Named/boolean arguments (no values)
4. Arguments requiring values
5. Multi-value arguments
6. Repeatable arguments
7. Tags
8. Positional arguments

Example with no positional arguments:

```
az vm create --resource-group myResourceGroup --name myVM --no-wait --image WindowsWhatever
```

Example with positional arguments:

```
az acr build --registry myregistry --image helloacrbuild:v1 ~/myappcode
```

## Variables

If possible, avoid reusing shell variables across multiple code blocks.

The reader may complete article steps in different sessions. Using variables across code blocks may cause errors (and frustration) in this situation, if they're not set correctly. If you must carry the variables across steps, make it clear that variables are reused in later steps.

## Format for Bash shell

Format your Azure CLI code blocks for the [Bash shell](#). For a good tutorial on the Bash shell, see the [Bash Beginner's Guide](#). The most common features of `bash` that you'll need to use in writing samples for the Azure CLI are:

- Line continuation character: `\`
- Variable set: `MYVAR=myvalue`
  - Don't use the `export` keyword when setting a variable. `export` can have unintended consequences.
- Variable get: `$MYVAR`
- Capture command output with `$(...)`: `MYVAR=$(execute-command)`
  - You may also capture command output with ``...`` but the `$(...)` syntax is preferred.
- Escape shell-interpreted characters with `'`: `MYVAR='literal$'`
- Force evaluation of shell-interpreted characters with `"`: `MYVAR="prefix-$OTHERVAR"`

## Randomize passwords for new resources

If you're creating a resource that will have a password associated with it, don't use a hardcoded password. Checking any passwords into source control, even examples, is a security risk.

The following methods are recommended to generate a password:

- `openssl rand --base64` will generate a series of random characters suitable for a password. Make sure the number of characters generated is divisible by 3 (use at least 15.) For example:

```
NEW_PASSWORD=$(openssl rand --base64 15)
```

- Use the `/dev/urandom` source to generate a password of arbitrary length and characters. The resulting password will meet complexity requirements. See [StackOverflow: How to generate a random string](#) for details.
- `uuidgen` will generate a new [UUID](#). Example:

```
NEW_PASSWORD=$(uuidgen)
```

If you need to discuss credentials for Linux virtual machines or scale-set instances (for example, when using `az vm create` or `az vmss create`), don't use passwords. The standard for Linux is to use an SSH key and Azure will autogenerated the SSH key if you use the `--generate-ssh-keys` parameter.

## Avoid naming conflicts

Some Azure resources, like Azure Container Registry and Key Vault, have resources tied to domain names. Those resources must have a *universally unique* name. For that reason, use a random value as part of names when uniqueness is required. If you don't, scripts will fail to create a required resource if more than one person runs them. Randomness won't *prevent* conflict, but can *mitigate* it.

Use `$RANDOM` to add a random number to a name. For example:

```
NEW_ACR_NAME="myacr-$RANDOM"
```

You can also create a random string identifier from `/dev/urandom`. See [StackOverflow: How to generate a random string](#) for details.

## Line continuation

If the command length causes horizontal scrolling, use a line continuation character and indent each following line 4 spaces. Go for readability over strict adherence to this rule. The example below uses single-line statements for two of the arguments to maintain readability.

```
az container create \
    --resource-group $RES_GROUP \
    --name acr-tasks \
    --image $ACR_NAME.azurecr.io/helloacrtasks:v1 \
    --registry-login-server $ACR_NAME.azurecr.io \
    --registry-username $(az keyvault secret show --vault-name $AKV_NAME --name $ACR_NAME-pull-usr --query
    value -o tsv) \
    --registry-password $(az keyvault secret show --vault-name $AKV_NAME --name $ACR_NAME-pull-pwd --query
    value -o tsv) \
    --dns-name-label acr-tasks-$ACR_NAME \
    --query "{FQDN:ipAddress.fqdn}" \
    --output table
```

## Output

When you show output for a command, tag the output code block as `output` (after the backticks). The output must be in a separate block from the CLI commands.

### IMPORTANT

Don't display command output in an image. Images break accessibility by defeating screen readers and interfere with web search.

```
az vm create -name MyVm -resource-group MyResourceGroup --image UbuntuLTS --generate-ssh-keys
```

An output block for the previous example command:

```
{- Finished ..
  "fqdns": "",
  "id": "/subscriptions/896e1936-a2ce-4761-9c66-5e3cec0bbba1/resourceGroups/abc/providers/Microsoft.Compute/virtualMachines/MyVm",
  "location": "eastus2",
  "macAddress": "00-0D-3A-7C-08-B0",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "20.4* 19.249",
  "resourceGroup": "MyResourceGroup",
  "zones": ""
}
```

## Interactive code snippets (Try It)

The use of the **azurecli-interactive** code tag is under discussion as **Try It** actually opens Azure Cloud Shell without copying the code sample. Also, code samples with [line continuation](#) characters must first be copied to a text editor, modified, and then copied back to the console. It is better to clearly explain how to install the Azure CLI in the "Prepare your environment" section of the article. If needed, add a link back to prerequisites.

## Unsupported CLI commands when using Cloud Shell

Not every Azure CLI command is supported by Azure Cloud Shell. If you mix Cloud Shell-functional commands with commands that don't work in the Cloud Shell, you risk frustrating customers when only *some* commands work. Use the **azurecli** language tag for all code blocks to help avoid confusion. For example,

```
# This command requires running the docker daemon, which is not supported in Azure Cloud Shell.
az acr login -n myregistry
```

## See also

- [Azure CLI editor's checklist](#)

# Azure CLI editor's checklist

5/5/2021 • 2 minutes to read

This document is a summary of rules to apply when writing new or updating existing Azure CLI articles. Use the links in each checklist item to go to [Azure CLI article guidelines](#) for additional detail and examples.

## Table of Contents

- New articles have been added
- Any change to a H1 has also been updated in the TOC.
- The `displayName` property has been added containing key CLI search words and reference names.
- Azure CLI has been added under the Reference node.

## Article metadata

- ms.custom contains the tag `devx-track-azurecli`

## Prerequisites

- One of three [Azure CLI prerequisite include files](#) have been added *before* the first CLI code block.
- The [required version](#) for the article's Azure CLI command group has been clearly provided.

## Code blocks

- A [link](#) to CLI reference content precedes the code block for each used command.
- The correct `azurecli` code tag has been used for reference syntax.
- The correct `output` code tag has been used for [command output](#), not a screen print.
- The use of `azurecli-interactive` and the consequent **Try It** button has been avoided.
- Multi-line commands use the [line continuation](#) character.

## Formatting

See [Guidelines for Azure CLI articles](#) for formatting examples.

- Full names, not alias abbreviations, have been used for parameter names.
- Backticks have been used for syntax elements that appear, inline, within a paragraph.
- Backticks have been used for reference names, --parameter names and values.
- Backticks have been used for file paths.
- Backticks have been used for non-clickable URLs.
- Bold has been used for command group names.

## Sample scripts

- All scripts have been *thoroughly tested*.
- If replicating a portal or PowerShell article, *identical object names* have been used.
- All CLI scripts are platform-agnostic, and code samples are formatted for [Bash shell](#).
- A !NOTE has been added if the article uses an Azure CLI [command not unsupported](#) in Azure Cloud Shell.
- [Variables](#) have been used correctly in code samples.

- [Parameter names](#) have not been abbreviated.
- [Argument order](#) has been verified.
- [Passwords](#) have not been hardcoded.
- The requirements for using [universally unique resource names](#) in code samples have been followed.
- Noise has been omitted, such as the console `azureuser@Azure:~$` line prefix. Each CLI sample begins with `az commandGroupName` as reflected in the [CLI reference docs](#).

## Final steps

- [Reference content](#) has not been copied from the CLI auto-generated docs.
- If the article has created Azure resources, a [Clean up resources](#) section has been provided at the end of the article.
- All Azure CLI build warnings have been fixed, even if inherited from another writer.

## Article examples

Every Azure service supported by the CLI has unique article requirements; however, here are three examples that will help you with a correct information flow.

- [Quickstart: Create and manage logic apps using the Azure CLI](#)
- [Quickstart: Create an Azure notification hub using the Azure CLI](#)
- [Quickstart: Define and Assign an Azure Blueprint with Azure CLI](#)

## See also

- [Azure CLI article guidelines](#)

# Azure PowerShell code conventions

4/16/2021 • 4 minutes to read

The following checklist helps ensure scripts and samples using Azure PowerShell are consistent. For more information on submitting standalone scripts and the pull request process, see [Azure tools script PR review process](#).

## Version

If your article requires a minimum version of Azure PowerShell, that version should be called out. Also mention any other requirements like PowerShell version or OS.

### IMPORTANT

If your article requires AzureRM, include the `includes/requirements-azurerm.md` file.

## Formatting syntax elements in a paragraph

PowerShell cmdlet and parameter names are [Pascal Cased](#). Always use the full Pascal Case name for cmdlets and parameters. Avoid using aliases unless you're specifically describing an alias. Property, parameter, and class names should be **bold**. Language keywords, cmdlet names, and variable references should be wrapped in backtick (`) characters. For example:

```
The following code assigns the output of `Get-ChildItem` to the `$files` variable...
```

The first mention of a cmdlet name should be a link to the cmdlet documentation.

For example:

```
The [Get-AzureRmVM](/powershell/module/azurerm.compute/get-azurermmvm) cmdlet uses the **Location** parameter to ...
```

### NOTE

Do not use backticks or bold inside the brackets of a hyperlink. This is an exception to the previous rule. For more information about linking to PowerShell content, see [Add links to articles](#).

When referring to a parameter by name, the name should be **bold**. When illustrating the use of a parameter with the hyphen prefix, the parameter should be wrapped in backticks. For example:

```
The parameter's name is Name, but it is typed as -Name when used on the command line as a parameter.
```

## Argument order

Arguments for an Azure PowerShell cmdlet should appear in the order defined by the cmdlet help. When a cmdlet has multiple ways to provide required arguments, follow the argument listing closest to the usage you're demonstrating. An example of a cmdlet with multiple ways to invoke it is [Connect-AzAccount](#).

# Variables

If possible, avoid reusing shell variables across multiple code blocks.

The reader may complete article steps in different sessions. Using variables across code blocks may cause errors (and frustration) in this situation, if they're not set correctly. If you must carry the variables across steps, make it clear that variables are reused in later steps.

## Randomize passwords for new resources

If you're creating a resource that will have a password associated with it, **don't** use a hardcoded password. Checking passwords into source control, even examples, is a security risk.

The following method is the recommended way to generate a password:

- `New-Guid` will generate a new [UUID](#). Example:

```
$NEW_PASSWORD=(New-Guid).Guid
```

Some services may have password complexity requirements that can't be met by a GUID (which uses lowercase letters, 0-9, and the `-` character.) In that case, use a static prefix or postfix.

```
COMPLEXITY_PASSWORD="PAS!0rD$((New-Guid).Guid)"
```

## Avoid naming conflicts

Some Azure resources, like Azure Container Registry and Key Vault, have resources tied to domain names. Those resources must have a *universally unique* name. For that reason, use a random value as part of names when uniqueness is required. If you don't, scripts will fail to create a required resource if more than one person runs them. Randomness won't *prevent* conflict, but can *mitigate* it.

Use `Get-Random` to add a random number to a name. For example:

```
$NEW_ACR_NAME="myacr-$(Get-Random)"
```

## Line continuation

Avoid using the PowerShell line continuation character if possible. PowerShell syntax allows for 'natural' line breaks after pipes, braces, commas, and within some syntax blocks. For long parameter blocks, use PowerShell [splatting](#) to improve the readability of your example.

Example using splatting (preferred):

```
$frontendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix "10.0.1.0/24"
$backendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix "10.0.2.0/24"
$parameters = @{
    Name = 'MyVirtualNetwork'
    ResourceGroupName = 'TestResourceGroup'
    Location = 'centralus'
    AddressPrefix '10.0.0.0/16'
    Subnet = $frontendSubnet, $backendSubnet
}
$vnet = New-AzureRmVirtualNetwork @parameters
```

If you must use line continuation characters, indent each following line 4 spaces.

Example using line continuation:

```
$frontendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name frontendSubnet -AddressPrefix "10.0.1.0/24"
$backendSubnet = New-AzureRmVirtualNetworkSubnetConfig -Name backendSubnet -AddressPrefix "10.0.2.0/24"
$vnet = New-AzureRmVirtualNetwork ` 
    -Name MyVirtualNetwork ` 
    -ResourceGroupName TestResourceGroup ` 
    -Location centralus ` 
    -AddressPrefix "10.0.0.0/16" ` 
    -Subnet $frontendSubnet, $backendSubnet
```

## Output

When showing example output for a command, tag the output code block as "Output" (after the backticks). For example:

```
Get-AzureRmVM -ResourceGroupName "ResourceGroup11" -Name "VirtualMachine07"
```

An output block for the previous example command:

ResourceGroupName	Name	Location	VmSize	OsType	NIC	ProvisioningState	Zone
ResourceGroup11	VirtualMachine07	centralus	Standard_B2ms	Windows	win2012ps3226	Succeeded	

## Interactive code snippets (Try It)

### When to use Try It

If *every* Azure PowerShell command in your article is supported by the Cloud Shell, tag your code blocks with `azurerpowershell-interactive` to add the [Try It](#) button to the snippets:

```
```azurerpowershell-interactive
Get-Command -Verb Get -Noun AzVM* -Module Az.Compute
```
```

### When NOT to use Try It

Don't use `azurerpowershell-interactive` if your article includes *any* Azure PowerShell commands that don't work in the Cloud Shell--use `powershell` only. For example, the [AzureAD module](#) cmdlets are unsupported:

```
Get-AzureADUser -Top 10
```

If you mix Cloud Shell-functional commands with commands that don't work in the Cloud Shell, you risk frustrating customers when only *some* commands work. Instead, stick with the `azurerpowershell` language tag for all code blocks to help avoid confusion.

## Update AzureRM to Az PowerShell module

The easiest way to translate the command names is by using [Enable-AzureRmAlias](#). First, install the [Az PowerShell module](#). Once you have the Az module installed, run the command named `Enable-AzureRmAlias`. Then you can run `Get-Alias` for the AzureRM command to translate it to the corresponding Az cmdlet name. The following example returns the Az cmdlet that maps to `Add-AzureRmAccount`.

```
Enable-AzureRmAlias
```

```
$Results = Get-Alias -Name Add-AzureRmAccount
if ($Results.ResolvedCommand) {
    $Results.ResolvedCommand
} else {
    $Results.Definition
}
```

# Technical focused reviews

4/16/2021 • 3 minutes to read

Technical focused reviews (TFRs) help ensure the technical accuracy and relevance of an article. In concert with [customer focused reviews](#) (CFRs) and their focus on customer intent and writing principles, TFRs verify that the article provides the best path to success given the intent of the content. Whereas a CFR validates your customer intent statement and that the article adheres to our [writing principles](#), a TFR allows for concentration on the technical aspects of the article.

Although you're not required to submit your articles for technical focused review, you're encouraged to do so. A TFR is recommended especially for technically complex articles or those that are performing poorly in customer satisfaction. Managers, leads, and content PMs can also recommend articles for a TFR.

## How do I sign up for a TFR?

Before you schedule a TFR, make sure you have time to implement any suggested changes. We recommend you schedule a TFR only for articles you can publish within five days. **TFRs are one hour long and focus on a single article.**

1. Select a date and time for your review. You do not need to reserve a calendar slot as you do with CFRs; pick any time that works for you.
2. Send an email requesting volunteer reviewers to [caiapexclfe@microsoft.com](mailto:caiapexclfe@microsoft.com).
  - Specify **TFR volunteer request** and your **service or article title** in the subject line
  - Include the article title and staging link in the body of the email.
  - Consider also inviting **product partners** (PMs and engineers), **Cloud Advocates**, and other subject matter experts (SMEs) to your TFR
3. Once you've identified your volunteers (4 or 5 is ideal), send them an Outlook calendar invite that corresponds with your **one-hour TFR timeslot**. Include in the invite:
  - A link to the article.
  - A brief statement identifying the [customer intent](#). Provide a clear customer intent statement when you send out the invitation to help your reviewers give you the most useful feedback.
  - A link to the [technical principles checklist](#).Send your Outlook invite and customer intent statement out **at least two business days in advance** to give the participants adequate time to prepare.
4. Ask one of your reviewers to be a facilitator for the TFR meeting. If no one in your group of reviewers is willing to facilitate, you can do so yourself.
  - Unlike CFRs, there are no official TFR facilitators. You can facilitate your own TFR, but it's recommended that you have one of your reviewers fill the role if they're willing.

## What can I expect?

In a TFR, you meet with 5 or 6 of your peers, product team partners, or other SMEs who review the technical aspects of your article and discuss ways it might be improved. We focus on one article per review, and each review typically takes one hour.

By conducting a TFR for your article, you can:

- Validate an article's technical consistency, accuracy, and relevance
- Confirm whether an article provides the most appropriate technical solution to a customer problem
- Help balance production-ready considerations with usability (e.g. security vs. brevity)
- Confirm an article's technical learning goals fit into the larger ecosystem

## Who should I invite?

Your review team should include a range of experience levels with your product, service, or language. It's helpful to gain perspectives from both subject matter experts (SMEs) and novices, as each can surface areas of improvement. Consider inviting not only fellow content developers, but also willing product team partners like PMs and engineers. If your article includes technologies or services other than your primary, try to include an SME for that technology, language, or service.

## I want to be a reviewer

Great! We'd love your input. Before the technical focused review meeting:

- Review the [technical principles checklist](#). The review is centered around these principles.
- Check out the author's intent statement before you review the article. Does it make sense to you? How effectively does the article support the intent?
- Plan on setting aside an hour to prepare and an hour for the review.
- Bring your comments to the meeting, so you can leave them with the author.

# The five writing principles

3/5/2021 • 5 minutes to read

## 1. Focus on customer intent

People are reading your article to find an answer to a question. In most cases, they're looking for how to do something. Figure out what that something is and focus the article on getting your reader to that answer.

- **Create a customer intent statement.** Customer intent identifies the job the customer is trying to do. It does not identify your reason for writing the article. Documenting the customer intent of your article helps keep the content focused and assists in portfolio planning. A list of customer intent statements gives you a better sense for the scope of your content before you begin writing.

Consider these best practices:

- Document the customer intent as a comment near the top of your article to reminder you (and others) of its focus.
- Use the agile user story format for creating customer intent statements: **As a < type of user >, I want < what? > so that < why? >**.

- **Organize the article so the most common intent is near the top and other information appears later.** Consider these best practices:

- For articles with a basic answer and more advanced options, put the basic information at the top and the advanced options afterward.
- If information covers a different intent, link to it rather than including it in your article.
- For articles with one procedure, put the procedure at the top followed by the conceptual information.

- **Make the most common task ridiculously easy to find.** Choose a technique based on what works best for your situation:

- Put the intent in the title.
- Put the intent in the first paragraph.
- Put the intent in a procedure preceding the table of contents.
- Put the intent in a section header.

### **Customer intent statements**

Here are a few examples of effective customer intent statements that follow the agile user story format.

- "As an IT admin currently completing tasks by clicking through the portal, I need to download and install the latest version of PowerShell, so I can automate those tasks."
- "As a developer, I need to build an application, so I can run parallel workloads using Batch."
- "As a device developer I need to know what Event Hubs does, so I can understand if it can help me manage the large amount of data I need to analyze."
- "As a business decision maker, I'm evaluating Azure Backup to determine whether it supports the backup and recovery scenarios that my company requires."

## 2. Use everyday words

Like all the writing guidance, word choice guidance depends on context—the article, the product, the reader, the feature. Choose words based on context and aim for "easy to understand" for that context.

- **Use everyday words.** Many words have a formal and everyday version. For example: *provide* and *give*

OR *obtain* and *get*. Formal versions tend to be longer and have more syllables. That doesn't make them better or more precise. Using them just makes the sentence longer. These words have nothing to do with the technical content of the article but have a huge impact on how the article sounds.

- **Use a technical word if it's the right word.** In an article about AD FS, the phrase *server farm* is perfectly appropriate. In an article about AD domains, the word *domain* is a good one to use. Understand your audience and use their everyday technical words.
- **Use inclusive language.** Microsoft technology reaches every part of the globe, so it's critical that all our communications are inclusive and diverse. Make sure you follow [inclusive language guidelines](#) in your writing.
- **Emphasize benefits, not technology.** When you choose words that focus on what the reader wants to do, the content is easier to understand. That is, the benefit is easier to understand than the underlying technology.

| WORDS EMPHASIZING THE BENEFIT  | WORDS EMPHASIZING THE UNDERLYING TECHNOLOGY   |
|--|---|
| To read email on your phone...   | To enable mobile access by using Exchange ActiveSync...                                   |
| Do you want to sign in once to all of the services in Office 365?                      | Do you want to deploy AD FS to your server farm in order to use federated identity?       |
| Let people outside your company access shared documents in a security-enhanced manner. | Control access permissions for external users to your SharePoint Online site collections. |

- **Read the article aloud.** Does it sound natural? Aim to sound like a coworker talking to another professional.
- **Recommended from the Microsoft Writing Style Guide:**
  - [Use contractions](#)
  - [A-Z word list and term collections](#)
  - [Security, safety, and privacy terms](#)
  - [Cloud computing terms](#)

### 3. Write concisely

Customers prefer articles with fewer words:

- **Try to cut the word count by half.**
  - Introductions, conceptual information, and notes are good places to reduce wordiness.
  - Sentences using active voice instead of passive are usually shorter, and easier to understand.
- **Use everyday words to help with conciseness.** Formal words lead to formal sentences.
- **Read the article aloud.** If you're out of breath, the article is too wordy.

### 4. Make the article easy to scan

People don't read our help articles like novels, from the first word to the last. They scan our articles for answers as if they are looking up a name in the phone book. Make articles easy to scan so they can find what they're looking for:

- **People scan titles, procedures, headers, art, and links.** After they find roughly what they're looking for, they then read for details.
- **Art is fantastic for drawing attention.** Would a screenshot help the reader find or understand something

better?

- **Scan the article and see what jumps out.** Which procedure looks most important? Which intent seems to be the focus? Which tips are easy to find? Sometimes less important things get the focus and more important things are hidden. Can the wording, organization, and formatting be changed so the right information is emphasized?
- **Optimize for SEO.** Search engines look for clues to find help articles that match what our readers search for. And they look in many of the places people do: titles, headings, alt-text, links, as well as in the body, so make sure that you are using your search keywords in these areas. See [SEO: Basic techniques for good web writing](#) for guidance on optimizing your content.
- **Recommended from the Microsoft Writing Style Guide:**
  - [Scannable content](#)

## 5. Show empathy, or at least, don't be rude

- **Remember that it's not us versus them.** Readers object to instructions that imply that Microsoft knows better than they do, and who can blame them? Our goal is to empower readers to solve problems, not to convince people that we're smarter than they are. Try to rephrase sentences that use phrases like "you must" and "you are required." Focus instead on the benefit of doing what will most likely solve the problem.
- **When you know that users are frustrated, consider acknowledging that.** Here are some places that apply:
  - An article that discusses how to resolve a perplexing error
  - A community forum post that's a response to a thread of yelling customers
  - A task that will likely take longer than users expect. Examples are deploying a server, deleting resources, and creating an image. Consider language like this: "Keep in mind that deleting these resources may take up to..."
- **Recommended from the Microsoft Writing Style Guide:**
  - [Use certain words carefully](#)

# Writing principles checklist

5/21/2021 • 2 minutes to read

Ask yourself the following questions to help determine if your article aligns with Content & Learning writing principles. You can also use this checklist to prepare for your CFR.

## Customer Intent

Before you begin writing, clearly determine who the customer is and what task they are trying to do. Then, write your article to help the customer do that task.

- What does the customer really want from this article?
- Can you capture that intent in the following format? As a < type of user >, I want < what? > so that < why? >.
- Did you [choose the correct content type](#) for the intent?

## Use everyday words

Use the words your customers use. Be less formal but not less technical.

- What keywords would the customer search for that would take them directly to this article?
- Does the page title, headline (H1), and introduction tell the customer they're in the right place?
- Does the rest of the article use words that your audience uses?
- Have you removed unnecessary Microsoft jargon?
- Does the article sound too formal when you read it out loud?
- Have you followed [inclusive language](#) guidelines?

## Write concisely

Keep your article focused on a single customer intent. Don't waste words. Keep sentences short and concise.

- Have you documented only what is necessary to achieve the intent?
- Is there any unnecessary information getting in the way?
- Does the article use locators (>) for "click, click, click" instructions?

## Easy to scan

Put the most important things first. Use sections to chunk long procedures into more manageable groups of steps.

- Is the article too long to allow for scanning?
- Are the most important things first in the article?
- Do the section headers lead the customer through the article?
- Does the article use tables, lists, and bolding?
- Does the article use art to guide the user through the content?

## Show empathy

Make sure the article focuses on what matters to customers; don't just give a technical lecture. Use a supportive tone.

- Does the article read like we are their partner?
- Does the article address the customer's point of view and what matters to them?
- If the situation is frustrating, does the article recognize it?

# Writing principles job aid

5/26/2021 • 2 minutes to read

| WRITING PRINCIPLES | CONSIDERATIONS  | INTENT   |
|--------------------|---|--|
| Use everyday words | <ul style="list-style-type: none"><li>Word choice is an important part of voice.</li><li>Be less formal, not less technical.</li><li>Use natural language. Don't use cute or casual language.</li><li>Imagine you're talking to a peer.</li><li>Use technical words when they're the right words.</li><li>Use words that your audience uses.</li><li>Explain new technical concepts using real-life examples.</li></ul> | <ul style="list-style-type: none"><li>Everyday words express the intent the way a reader does</li><li>Use keywords in the title and description</li><li>Use keywords in the section headers</li></ul>  |
| Write concisely    | <ul style="list-style-type: none"><li>Use everyday words.</li><li>Use locators (&gt;) for "select, select, select" instructions.</li><li>Don't repeat words that are in the UI.</li><li>Read the article out loud. Do you stumble over some words or sentences? Are you out of breath? If so, simplify.</li><li>Consider using art.</li><li>Delete words that aren't essential.</li></ul>                               | <ul style="list-style-type: none"><li>It's easier to be concise when you're focused on one thing</li><li>Use intent data to suggest what you can delete or move elsewhere</li></ul>  |
| Easy to scan       | <ul style="list-style-type: none"><li>Write concisely.</li><li>Consider using art.</li><li>Use tables, lists, and bolding.</li><li>Put your section headers to work.</li><li>Take the birds'-eye view.</li><li>Put the most important thing first.</li></ul>  | <ul style="list-style-type: none"><li>The article's main intent is impossible to miss</li><li>The title and description tell me I'm in the right place</li><li>The section headers lead me through the article</li><li>There's no unnecessary information getting in my way</li><li>The article is organized so I can find my answer</li></ul> |

| WRITING PRINCIPLES | CONSIDERATIONS  | INTENT   |
|--------------------|---|--|
| Show empathy       | <ul style="list-style-type: none"> <li>● Be a partner, not a judge.</li> <li>● Talk like you're helping a coworker.</li> <li>● Don't imply the problem is their fault. <ul style="list-style-type: none"> <li>○ If the situation is genuinely frustrating, acknowledge it.</li> <li>○ You don't have to be their buddy, but don't be a jerk.</li> </ul> </li> <li>● Leave disclaimers to the lawyers</li> </ul> | <ul style="list-style-type: none"> <li>● The tone of the article supports the customer</li> <li>● The article addresses the customer's point of view and what matters to them</li> </ul> |

# Everyday word list

5/26/2021 • 2 minutes to read

Use this list to help you replace formal words with everyday words.

| FORMAL           | EVERYDAY  |
|------------------|---|
| access           | use   |
| acquire          | get   |
| administrator    | admin   |
| allows           | lets  |
| alternate        | switch  |
| as follows       | here are (here is)  |
| as               | because   |
| broad categories | types   |
| Click            | Select  |
| Click here       | <p>Don't use. For links to web pages, use something like<br/>For more information, see [title of page as link]<br/>(link.md)</p> <p>.</p> |
| Click on         | Select <b>UX element</b> .  |
| Click the X tab  | Select <b>X</b> .   |
| complete         | finish  |
| configure        | set up  |
| contains         | has   |
| correct          | fix   |
| determine        | check   |
| displays         | shows   |
| do the following | follow these steps  |
| drag and drop    | drag  |

| FORMAL                       | EVERYDAY   |
|------------------------------|--|
| due to                       | because  |
| elect                        | choose   |
| e-mail                       | email (ok to use as a verb)                          |
| enable                       | let  |
| enables                      | lets   |
| enter a maximum of           | type up to   |
| execute                      | run  |
| forward                      | next   |
| halt                         | pause, stop  |
| however                      | but  |
| illegal                      | not valid, can't be (unless in a legal context)      |
| impact, impacting, impactful | affect (verb), effect (noun), affecting, effective   |
| it's recommended             | we recommend   |
| log in                       | sign in  |
| mail                         | email  |
| media                        | music, video, photo (be specific)                    |
| modify                       | change   |
| More information             | <a href="#">More about [specific topic](link.md)</a> |
| navigate                     | go   |
| obtain                       | get  |
| operation                    | action, task   |
| perform                      | do   |
| provision, provisioning      | set up, setting up                                   |
| purchase                     | buy  |
| repair                       | fix  |

| FORMAL         | EVERYDAY                                       |
|----------------|--|
| resolve        | fix  |
| such as        | like   |
| suspend        | pause, stop                                    |
| synchronize    | sync   |
| terminate      | end, exit                                      |
| type or select | enter, put (use a single word)                 |
| unavailable    | dimmed, grayed out only to describe appearance |
| understand     | learn  |
| various        | different                                      |
| verify         | check  |
| visit          | go to  |
| won't work     | doesn't work                                   |
| workload       | service, app                                   |
| you should     | we recommend (use sparingly or write around)   |

# Customer Focused Reviews

8/26/2019 • 2 minutes to read

Customer focused reviews are no longer supported for C+L. The [MS Voice principles](#) still apply and guidance is still available to help ensure your content aligns with that approach. You can still use the guidance to lead your own peer reviews using the voice principles.

# SDK & reference content guide

11/2/2020 • 2 minutes to read

Welcome to the [docs.microsoft.com](#) SDK and reference content guide!

## Get started

- If you're on an Azure Service team, you should start with the Azure SDK onboarding process at [aka.ms/joinTheSDK](#).
- [API reference documentation onboarding guide](#) contains everything you need to know to get your API reference live on docs.microsoft.com.
- [Reference documentation overview](#) orients you to the collection of GitHub repositories used for API reference and the directories within those repositories, used for organizing reference content creation and generation.
- [Client library documentation guide](#) helps you document your library, both API reference and supporting documentation like its repository README and package distribution (NuGet, PyPi, npm, Maven) landing page.
- [Understanding Swagger](#) - Many Azure Service teams are using Swagger files to describe their service interfaces (the REST APIs). This creates the opportunity to couple basic service documentation to the service description process.

## How to

- [Document your Azure client library](#)
- [Write an Azure client library Quickstart](#)
- [Write .NET docs](#)
- [Write CLI docs](#)
- [Write Powershell docs](#)
- [Tag code examples](#)

## Resources

- Template: [Library reference index](#)
- [Reference style guide](#)
- [Azure SDK onboarding](#)

# Azure reference documentation - overview

11/2/2020 • 6 minutes to read

We are introducing a new reference documentation process that will convert your Swagger API element descriptions and code comments into reference documentation on:

<https://docs.microsoft.com/<language>/api/<your-service-here>>

## Why a consistent model for API reference

Managing large portfolios of reference content is a big drain on our resources, resources we want to focus on broader feature coverage and key customer scenarios.

The current reference doc model results in docs that are:

- Well behind the time of deployment
- Often out-of-date
- Inaccurate if unreviewed; reviews sap the time of content AND engineering
- Difficult to maintain with too much coordination required for fast iteration
- Lacking in details (or incomplete entirely)

Further, in Agile and CD processes, we are still downstream; reference content must be maintained in-line with any updates or deployments. The folks best-suited to perform maintenance at this pace are the folks who created the API code, the service or product devs such that what customers see in public repos will be reflected in the docs.

API reference is a critical deliverable and, it has scaled, and will continue to scale, beyond our resources. Therefore we must ensure products and services with developer aspects ship with high quality API ref, using:

- Telemetry - ID key APIs and customer pain points.
- Skills (- ensure that the most critical APIs have richer experiences.
- Customer focus - make sure the dev-authored content accrues well to our key scenarios and stories – drive focus on the overall content experience, not just the ref itself.

To be successful, these docs must use code element comments and descriptions that are part of your workflow and:

- Are kept up-to-date for each version.
- Are easily discovered and interpreted by doc tool agents.
- Are well-written and will read clearly on a public web page.
- Are accurate in a way that only happens when the code creator writes them directly.

This new approach to API ref is a chosen leadership model for our org so, bring your skills and experience to the broader table!

## Developers own API reference documentation

Developers own and write Swagger 2.0 specifications for REST APIs and doc-ready comments for client SDKs. The “description” strings in those specs (currently azure-specs-rest-api) provide the basis for the REST API documentation. For client SDKs, we use:

- /// comments for .NET and Java SDK docs.

- help.py files for Python SDK and Azure CLI 2.0 docs.
- Some teams populate these comments from Swagger as part of AutoRest conversion!

## OPS resources

For everything you need to know to get your API docs live, visit the [API Documentation Onboarding guide!](#)

For more on content types supported by docs, see [What type of documentation do we host in docs.microsoft.com](#) and for details about general onboarding to OPS (Open Publishing Services), see [Onboarding to Open Publishing](#).

## How docs get built

The documentation test and build process runs as a continuous integration (CI) process (AppVeyor + doc generator tool) to:

- Push the Swagger JSON into the REST API doc repo.
- Convert the comments/help into YAML and push them to the client SDK doc repos.

Open Publishing (OP) runs DocFX over the Swagger/YAML sources and publishes the generated content to [docs.microsoft.com](#).

Example docs:

- [REST](#)
- [.NET SDK](#)
- [Java SDK](#)
- [PowerShell](#)
- [Python CLI](#)

## Roles and responsibilities

For Azure Service reference creation, the roles for the service developer and content writer are changing to a partnership model where the developers provide the initial reference content and the writers edit for quality. This change is being driven by Scott Guthrie and has the full support of engineering and documentation.

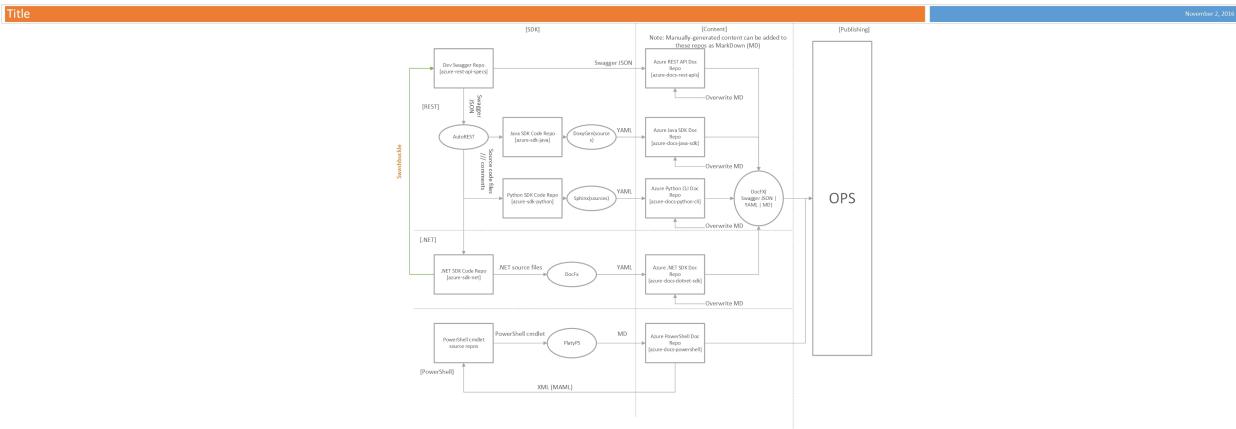
### Developers

- Are responsible for the baseline content and quality of all Azure ref content.
- Own the Swagger 2.0-based specs.
- Approve the PRs for Swagger definition changes pushed to `azure-specs-rest-api`.
- The long-term goal is to have this process be a reliable and automated "CD" process with minimal-to-no content team involvement.

### Content team

- Work in the intermediate Azure ref "doc" repos.
- Creates overwrite files to enhance select docs due to current limitations in the expressiveness of the Swagger JSON format (this will change over time).
- Reviews Swagger specs and ensures professional quality and consistency.
- Conducts training (or works hands-on) to drive best practices for documentation.

## Automation work flow for [docs.microsoft.com](#)



Page 1

Developers create content by adding comments for the baseline (F-level) reference content, per code element (type, class, interface, method, and others):

- Abstract (summary)
- Operation (description)
- Parameters (description)
- Return Values or Responses
- Error Codes and Exceptions
- (optional) Links to samples and docs (externalDocs)

Developers complete this step by writing comments or populating doc fields as follows:

- **REST API ref** are authored in Swagger JSON (akin to a spec) for the baseline reference content attributes:
  - “description” field
  - Note: “Server-side” REST APIs (usually the base http model) must be authored using the correct context. Likewise, cross-platform APIs that provide client-side calls to REST APIs must be authored with the client perspective in mind. More on this later!
- **.NET API refs** are sourced from dev authored triple-slash `/// comments` in the source code.
- **PowerShell refs** are authored using the *Cmdlet Designer* tool and extracted via **PlatyPS** for publishing.

Participate in the code review and/or review the code to ensure clear style; log bugs or Github *issues* as needed.

Then, as a continuous integration (CI) task, run by Visual Studio China (VSC), processes the sources and outputs the results to an **intermediate** private “doc” repo.

- REST (HTTP) – Swagger JSON plus DocFX.json and ToC.yml files.
- Cross-platform client SDKs – YAML docs for each type (DocFX.json and ToC.yml files).
- .NET SDKs - YAML docs for each type (DocFX.json and ToC.yml files).
- PowerShell APIs – MarkDown files for each cmdlet (DocFX.json and ToC.yml files).

### IMPORTANT

Never edit these files directly without approval. Approval will only be granted in extreme emergencies.

The diagram above shows how these various language sources flow through our system to complete as API ref content on [docs.microsoft.com](https://docs.microsoft.com).

# The Content Developer's role

Developers are the API reference writer and are responsible for API reference quality control.

There are four situations where you may need to work in corresponding doc repo:

- You need to update the index.md pages the ToC.md files. This update process may be automated in the future.
- We have an immediate need to hand-author ref content (MarkDown only) as part of a migration effort.
- You need to significantly improve a key reference page with remarks and code examples using "overwrite" MD files. We expect this to be an edge case covering, at most, 5% of the APIs.
- You need to create an "overwrite" file for base code element documentation because there is a high priority issue with published content. You'll delete this work when the devs push a fix to the docs in their sources.

## Doc repo rules

- Never edit anything but MD content directly.
- Overwrites should be used very sparingly and only when justified by customer data/feedback or a priority publishing issue. Overwrite files are found under `https://github.com/Azure/azure-docs-<model>-<lang-or-framework>-<some optional discriminant tag>` and named like
  - azure-docs-rest-apis (a slight naming convention exception, grandfathered)
  - azure-docs-sdk-java
  - azure-docs-sdk-dotnet
  - azure-docs-powershell
  - azure-docs-cli-python
  - azure-docs-misc (catchall content repo for as-yet difficult to classify APIs, MD-only)
- For example:
  - azure-docs-rest-apis (a slight naming convention exception, grandfathered)
  - azure-docs-sdk-java
  - azure-docs-sdk-dotnet
  - azure-docs-powershell
  - azure-docs-cli-python
  - azure-docs-misc (catchall content repo for as-yet difficult to classify APIs, MD-only)
- The /live branch is the publishing branch. Only content that is ready for publication should be pushed to it.

When you're ready to get started with your reference authoring in Github, our Contributors Guide has all you need.

## Files and directories

Each repo has a set of predefined directories and files that support the OPS documentation build process.

- **breadcrumb** - directory
- **docs-ref-autogen** - a directory of auto generated JSON files from OPS build tools
- **docs-ref-conceptual** - a directory of conceptual topics that augment the reference
- **docs-ref-overwrite** - a directory of overwrite files used to replace description fields in the swagger
- **index.md** - starting place for rendering of content
- **TOC.md** - table of contents for docset

## Edit this page!

This page is a work in progress. Want to help improve it? Click the Edit icon and submit your changes!

# Frequently asked questions (FAQ) about reference content

11/2/2020 • 10 minutes to read

This article addresses common questions about contributing to and understanding reference content on docs.microsoft.com. For detailed information and requirements for publishing a new API reference, see [API Documentation Onboarding](#) in the onboarding guide.

## Getting started

### How do I get my reference content on docs?

The first step is to submit an onboarding request: [aka.ms/publish-on-docs](https://aka.ms/publish-on-docs). From there, the Onboarding team and Docs Dev Experience PM team will respond.

One exception to this process is Azure PowerShell reference content. In this scenario, engineering teams contribute to an existing managed repo and the new reference documentation will be published live according to the schedule for that pipeline. For more information, see [Contributing Azure PowerShell content](#) in the Onboarding Guide.

### How do I get help with reference docs?

For support and questions, open a ticket with [SiteHelp](#). This allows us to track the frequency and type of issues with reference documentation generation and publishing.

### What languages do you support for autogenerated reference documentation?

The most up-to-date info on our supported platforms for automation is in the [onboarding guide](#). We autogenerate reference documentation for the following languages:

- .NET
- Python
- JavaScript
- Java
- REST
- PowerShell
- Azure CLI

### My language is not supported for autogeneration. Can I publish my reference content?

If your API/SDK library uses an unsupported platform, you can publish a [conceptual reference](#) only, meaning the source content must be manually written in Markdown.

### Where can customers find my reference documentation?

The main language browsers, which include search, are:

- [REST API browser](#)
- [Java API browser](#)
- [.NET API browser](#)
- [JavaScript API browser](#)
- [Python API browser](#)

Some API documentation is presented together logically, such as [Azure PowerShell](#) documentation and [Azure](#)

[CLI](#) documentation, which is organized by service (see under **Reference** in the TOC).

Some reference documentation is freestanding. We will help you decide where your documentation belongs.

## Repo ownership and administration

### Do I need a separate repository to host my code reference?

Not always. If you are an Azure product, we will build your content into existing repos. If not Azure, we will create a new repo for you. When applicable, your new repo might be created in the GitHub organization for that language if there is one (for example, <https://github.com/dotnet>).

This is something product stakeholders and the onboarding team will discuss/decide during initial intake of your onboarding request. You can find detailed guidelines and process info in the onboarding guide for each language.

### Does the Docs Dev Experience PM team handle admin tasks for reference content repos?

- Centrally managed repos: We provide admin support for managed reference content from autogeneration when you submit [SiteHelp tickets](#).
- Self-managed repos: For self-managed repos, we onboard and train your team and then hand the repo over to you and you are expected to manage your own ongoing builds, updates, etc. However, you can still request admin-level support via [SiteHelp tickets](#).

### Who manages the builds for my reference documentation?

When a new package is uploaded, we use continuous integration to automatically build and update your reference documentation. Later builds depend on the repo. For Azure reference content, you must request a build for minor changes outside of normal build times. For self-managed repos, you are expected to understand the publication process and are responsible for all builds.

### Why are some reference docs editable and not others?

This decision is up to the product team. Some product teams lockdown their docs, others don't, and some teams overlook that setting altogether!

If your content is autogenerated from an open-source repo, we can set up a 1:1 mapping so that any updates are reflected in the docs. If a customer submits a PR against your content and you accept the change, you must then manually update the source code.

If your code is hosted in a private repo, or the package is configured some other way that prevents a 1:1 mapping, you will not have the option to make your reference editable.

### How do I find the current owner of a repository?

Find the repo in the [Microsoft Open Source site](#) (not GitHub).

## Table of Contents (TOC)

### How do I fix my entry in the table of contents?

The table of contents (TOC) is generated as part of the reference content generation process. If your reference content is part of the four core languages (C#, Python, JavaScript, Java) and in the Azure service repos (not your own service repo), you control the TOC entry from the `docs-ref-mapping` directory files.

| LANGUAGE | UNIFIED TOC LOCATION |
|----------|----------------------|
| Java     | <a href="#">Java</a> |
| C#       | <a href="#">C#</a>   |

| LANGUAGE | UNIFIED TOC LOCATION |
|----------|----------------------|
| JS/TS    | JavaScript           |
| Python   | Python               |

If your reference content appears in the **Other** category at *the bottom of the TOC*, you have one or two problems:

- your reference content entry's `children` node has mistyped the package name
- your reference content entry is missing

Your reference content entry should include the following properties, using the Anomaly Detector service as an example entry:

```
- name: Anomaly Detector
uid: azure.python.sdk.landingPage.services.AnomalyDetector
href: ~/docs-ref-services/ai-anomalydetector-readme.md
children:
- 'azure-ai-anomalydetector'
```

To fix your entry, create a PR with a single file change to the appropriate `reference-X.yml` file. Use your previous generation and publishing process with the PR to get the change published to the live reference content.

If you have made this change but still see your entry in the wrong place in the TOC, create a [SiteHelp ticket](#).

## Publishing and releases

### Does the Docs Dev Experience PM team have a release schedule outlined?

We only schedule releases across groups for large events. For self-managed releases, we can turn around your reference content in a few days, but *only if nothing is broken*.

As a best practice, we recommend you contact us at least a week in advance before your release.

### Is reference content published on a regular schedule like larger conceptual docs (Azure, SQL, etc.)?

Main reference repos for Azure build 4-5 times a week. So, if you make a change in the SDK repos/source code, that change will not go live automatically. Instead, it goes live with the next build. If you need your content to go live sooner, request an out-of-band publish via [aka.ms/publish-on-docs](#) or the [Docs Support Teams channel](#)

For self-managed reference repos, each team defines their own schedule and which tasks are automated vs. manual. This can cause a problem for product team members who don't know exactly when their changes will go live, so it is important to train your team and communicate your builds.

For information or training on how to setup an automated build schedule, post to the [Docs Support Teams channel](#).

### How can I test my reference content before it's fully built?

A new package will not go live until your release date, so you can load, build, and interact with your content in advance. Our team reviews all content before go-live as part of our standard onboarding process.

And yes, it is possible to build preview packages. We can publish to monikers that are not released so you can review your content before it is live. This is for package-based systems. If you are .NET and don't have a NuGet package, there is another workaround in XML. It's not ideal, but it can be done.

If you have **embargoed** reference content, there might be limits on what we can build/preview for you. We will explain these options/restrictions as we handle your scenario.

## **My reference content is not rendering correctly on the review and/or main site. What do I do?**

You can post to the [Docs Support Teams channel](#) for input on your backend content. If you cannot figure out the issue, request help in the [SiteHelp portal](#).

## **Conceptual content**

*Conceptual content* refers to markdown articles such as overviews and tutorials that you write to help customers use your API.

### **When would my reference documentation live in the TOC of a larger guide vs. independently?**

If your API is for an Azure service, you will most likely be added to existing Azure-specific guides. If not Azure, you might be added as a new node in an existing guide (such as the [.NET guide](#), where your source files would land in a new folder of the [existing repo](#)). Or, your API will live as a freestanding guide with your own self-managed, independent repo. Whether your code is open source or not might also affect the location of your reference docs.

This decision is something product stakeholders and the onboarding team will determine during initial intake of your [onboarding request](#).

### **Can I host conceptual articles in the same repo as my reference documentation?**

In most cases, you should not add conceptual files to your reference repository. Instead, you'll host all conceptual content in the same repository as your product documentation. If you don't have associated product documentation, we will create a separate repo to host your conceptual content.

We host this content separately for better content governance and to avoid unnecessary problems each time you publish updates to your code. Also, reference documentation repos don't have the same quality automation that is available for conceptual repos.

There are limited exceptions that we identify during onboarding. For example, SDK libraries have a *library index* and *library overviews* in their reference documentation. For details on documenting an SDK library, see [Document your Azure SDK client library](#). While the guidance in that article is focused on Azure libraries, the index and overview articles apply to all library reference.

### **If my conceptual articles are hosted separately from my reference, how will customers find them?**

We can use a fusion TOC to display content from both repos (reference and conceptual) so they appear as a single guide. You can read more about [fusion TOCs](#) in the New Hope guide.

### **Why do some reference docs have a conceptual article section and others don't?**

Product teams add conceptual articles to their reference docs (usually with a [fusion TOC](#)) for one of two reasons:

1. The reference is not specific to a product or service, so there is no obvious conceptual doc set to add these articles to. Examples:
  - [Microsoft PROSE Code Accelerator SDK](#): Generates python code for data preparation tasks and can apply to multiple products and scenarios. These conceptual articles talk about how to use the SDK to work with Python data (for example, [Fix data types](#)).
  - [ML.NET Guide](#): Part of the larger .NET guide, though could be argued to live independently.
2. The conceptual articles are specific to the API/SDK. Often an Overview page, and sometimes more. Examples:
  - [Install the Data Prep SDK](#)
  - [Multi-cloud use of the Azure SDK for Python](#)
  - [Overview of Azure PowerShell](#)
  - [Service Bus Java libraries](#) (Overview)

When an API/SDK integrates with a Microsoft product/service that has a full conceptual guide, product teams should add conceptual articles (tutorials, how-tos, etc.) to that product's guide (vs. or in addition to the reference

TOC). Examples:

- Machine Learning service docs tutorial: [Prepare data for regression modeling](#). This tutorial is also linked to from the data prep ref space. However, the part 2 follow-up tutorial is only in the ML docs, as it does not involve the SDK.
- Machine Learning service crosslink: [ML.NET mention](#) in a Resources topic on Microsoft ML products
- Resource Manager docs how-to: [Deploy resources with RM templates and Azure PowerShell](#). This article is also linked to from the Azure PowerShell reference.
- Service Bus Messaging docs quickstart: [How to use Service Bus queues with Java](#). This service has samples, how-tos, quickstarts for different languages.

Some API docs have no conceptual articles, for example the [Power BI REST APIs](#).

**I see a conceptual section in this reference, but I can't edit it so I can't see what repo it is (in GitHub). How do I find the source repo?**

You can find the GitHub URL in the page source.

1. In the Chrome browser, right-click on a webpage and select **View page source**.
2. Look for the `original_content_git_url` attribute and you will find the repo URL appended with the page file path.

For example, if you view the page source for the [Data Prep SDK overview](#), you will find this line item:

```
<meta name="original_content_git_url" content="https://github.com/MicrosoftDocs/MachineLearning-Python-pr/blob/live/DataPrep/docs-ref-conceptual/intro.md">
```

MicrosoftDocs is the GitHub organization, MachineLearning-Python-pr is the repository, and intro.md is the source file for the overview article.

## Have a question?

Post to the [Docs Support](#) or [Docs Contributor Program](#) Teams channel for help. We will continue to update this article with frequently asked questions.

## Improve this article

This page is a work in progress. Want to help improve it? Select the **Edit** icon and submit your changes!

# Swagger interface specification

11/2/2020 • 2 minutes to read

"The Swagger specification is a powerful definition format to describe RESTful APIs. The Swagger specification creates a RESTful interface for easily developing and consuming an API by effectively mapping all the resources and operations associated with it. It's easy-to-learn, language agnostic, and both human and machine readable."  
- from [Swagger.io/specification](#)

Swagger has been used to define Azure Service interfaces for several years. These interface definitions are meant to be programming language neutral. Our goal was to seize the opportunity to link service documentation to the service definition phase of an Azure Service team's design process. We have partnered with the Azure SDK team to coordinate a tools driven documentation generation system which is based on the Swagger 2.0 definition of an Azure Service.

By including summaries and descriptions for all of the operations provided by an Azure Service, we can now produce generated documentation that remains in sync with the definition of that service.

Having the basic documentation coupled to the API definition means that it would be incorrect to have, for example, C#, Java or REST specific content in any of the swagger description fields. The content in these fields flows downstream into the various generated language interfaces that are outputs of our Azure Ref docs process.

## Who owns swagger interface definitions

Creating and maintaining swagger interface file and interface element descriptions is owned by the Azure Service team that owns the Azure Service. In order to bring the descriptions up to publishing quality, some writer assistance still may be needed. Since swagger is meant to describe the interface in a language neutral form, the descriptions must be edited to support this language neutrality. For more information on the details of this role see, the *Roles and responsibilities* section of [Overview](#)

## Working with an Azure Service team

The initial interface description content for all of the elements defined in a swagger file should be written by the owning Azure Service team. However, the service teams may not always adhere to our documentation standards for completeness and consistency. Responsibility can shift to the developer docs writer to work with the service team and edit the initial interface element descriptions to bring them up to our publishing standards.

The process of updating the documentation content for an Azure Service team's swagger file is an interactive one, done through the Github collaborative authoring processes. Once the swagger doc content has been updated, the swagger file should be signed off by the Azure Service developer and merged into the master branch for that service. You will be working with both the service team's developer and a member of the Azure SDK team, the owners of the Azure Service swagger standards and creators of a suite of [Azure Service swagger validation tools](#).

For more information on delivering developer experiences through Azure client libraries and tools, see [Deliver Awesome Programming Experiences for Azure!](#).

## Swagger extensions

By creating a few Microsoft specific extensions to the Swagger 2.0 standard we can insert additional information into the docs generation process.

- x-ms-extended-description - use to augment the swagger spec with API language specific guidance for the generated reference content. (TODO DOUGE - this extension may not be implemented due to other potentially better approaches - ex. Literal Swagger)
- x-ms-examples: use by the Azure Service to add sample code to the generated reference content.

## OPS Docs Resource

- [Migrating to Swagger-generated documentation](#)
- [Swagger spec cheat sheet](#)

# Document your Azure SDK client library

11/2/2020 • 6 minutes to read

Follow these steps to provide clear, complete, and discoverable documentation for your client library:

1. Document your code: [.NET](#) | [Java](#) | [JavaScript](#) | [Python](#)
2. Write the client library [README](#), publish on GitHub
3. Publish your package (NuGet, Maven, npm, PyPi) - use [README](#) as distro page
4. Request API reference onboarding to docs: <https://aka.ms/publish-on-docs/reference>
5. *Wait for API ref to go live, then...*
6. Publish [library index](#)
7. Publish [library overview](#) - use README as library overview
8. Write the client library [Quickstart](#)

Product team / Azure Developer Platform (ADP) team

Content team

## Documentation deliverables

There are several documentation deliverables for an Azure client library: the README (used in multiple places), a library index and overview, its API reference, a Quickstart, and "conceptual" documentation.

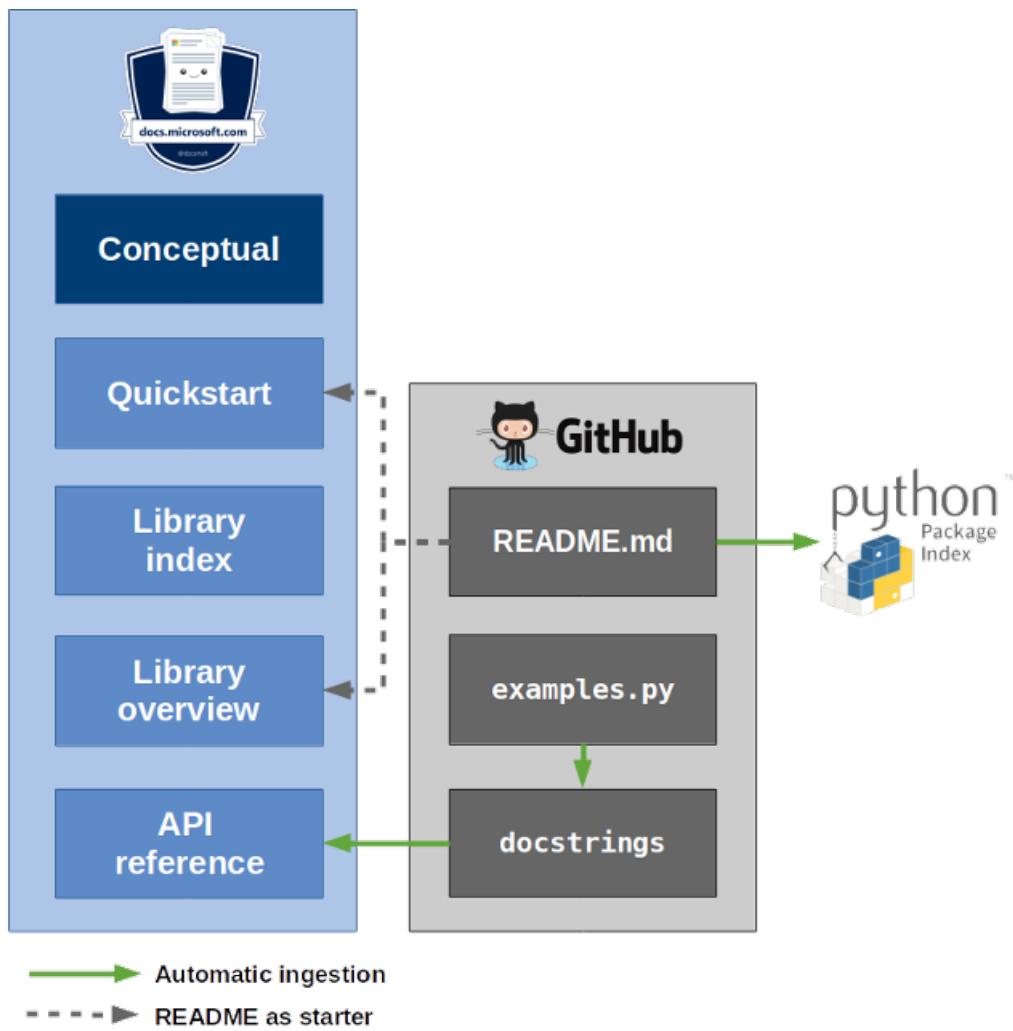
TYPE	DESCRIPTION	OWNER(S)	EXAMPLE(S)	TEMPLATE
README	<p><code>README.md</code> in the root of the library's repository or its directory within a repository, used for:</p> <ul style="list-style-type: none"><li>• Library README (GitHub)</li><li>• Package distro page (NuGet, PyPi, npm, Maven)</li><li>• Library overview (docs.microsoft.com)</li><li>• Starter for Quickstart (docs.microsoft.com)</li></ul>	Azure SDK team, product team	<a href="#">Cosmos DB client library for Python README</a>	<a href="#">README template</a>

Type	Description	Owner(s)	Example(s)	Template
API reference	API reference documentation for a library's types and their members. Auto-generated from the comments in the library source code. Includes code snippets ingested by and displayed in the published reference documentation.	Azure SDK team, product team	<ul style="list-style-type: none"> <li>• <a href="#">Service Bus Python reference</a></li> <li>• <a href="#">examples.py</a></li> </ul>	N/A
Library index	The library index is the landing page for a service's library reference on docs.microsoft.com. It lists and describes the libraries available for the service and links to their library overview pages.	Content developer, product PM	TODO	<a href="#">Library index template</a>
Library overview	The library overview is the landing page for a single library's reference on docs.microsoft.com. The library's README is reused for this.	Content developer, product PM	TODO	<a href="#">README template</a>
Quickstart	Quickstart that gets a developer started with the client library.	Content developer, product PM	TODO	<a href="#">Client library Quickstart</a>
Conceptual	Long-form technical documentation for the service. Concepts, quickstarts, tutorials, and how-tos.	Content developer, product PM, public	<a href="#">Cosmos DB documentation</a>	N/A

#### NOTE

The "Owner(s)" column identifies *typical* ownership for each type--we build the experience and ensure its quality **as one team**.

Here's an overview of how these doc types relate to each other for a Python library:



## README

The `README.md` is the entry point to your library. It's the welcome mat and front door, and should provide the quickest, easiest path for getting up and running. It should be as brief as possible but as complete as necessary, enabling developers to perform the most common operations with the least amount of friction.

- Template: [README template](#)
- Example: [Cosmos DB Python library README](#)

Author your README *once*, then **reuse it as**:

- Package distribution page (NuGet, PyPi, npm, Maven)
- Library overview
- Starting point for the library Quickstart

Include these sections in your README:

SECTION	DESCRIPTION
Introduction (no H2)	Appears directly under the title (H1) of your README. <b>Do not</b> use an "Introduction" or "Overview" heading (H2) for this section.

SECTION	DESCRIPTION
Getting started	Step-by-step instructions for obtaining and installing the package. Includes subsections: <ul style="list-style-type: none"> <li>• <b>Install the package</b> (required)</li> <li>• <b>Prerequisites</b> (required)</li> <li>• <b>Authenticate the client</b> (if applicable)</li> </ul>
Key concepts	Briefly explain the object model. Point out the most important and useful classes in the library, and briefly explain how those classes work together .
Examples	Code snippets and their descriptions for those operations that most developers will use. Include examples for operations that are complex or otherwise tricky to use.
Troubleshooting	Describe common errors and exceptions, how to "unpack" them (if necessary), and include guidance for graceful handling and recovery.
Next steps	Provide pointers to related libraries, documentation, or otherwise helpful content in other locations.

## API reference

API reference documentation is generated from the documentation in your code. For example, docstrings in Python and triple-slash ( `///` ) comments in C#. It enables the generation of the [API reference](#) that appears on docs.microsoft.com, and IntelliSense-like hover help in IDEs.

Follow the documentation best practices for the language in which your library is written. At a minimum, include the following:

- **All types, their members, and parameters are documented.** Provide typical usage instruction and include pointers to related members or help content.
- **Code snippets** are included for every major and "tricky" operation. The code for these snippets should be:
  - As short and simple as possible, yet as long as necessary to make it easy to use and follow. *Everyone* will copy+paste your snippets into their code.
  - Located alongside your library code (in the same repo).
  - Tested in an automated fashion.
  - Ingested into your code automatically via includes. For example, Python's Sphinx supports the `literalinclude` directive.
- **Return types** are fully documented. Not just its type, but how to use or interact with the returned thing.
- **Exceptions and errors** are fully documented. Your docs should answer questions like:
  - "What errors/exceptions are commonly encountered when using this method?"
  - "How do I gracefully handle and recover from this exception? Best ways for preventing it from occurring in the first place?"
- **Crosslink** between your reference docs, the README, and conceptual docs.
  - Provide links to types/members in the reference documentation from the README and conceptual docs, and vice-versa.
- Code documentation **validation tools** are available for most languages to help validate your in-code documentation--use them! For example, Python docstrings can be validated with [flake8-docstrings](#), [pydocstyle](#), and [docformatter](#).

**TIP**

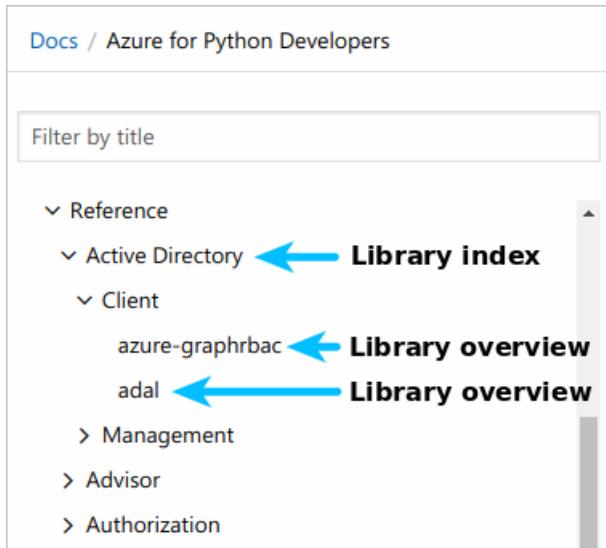
See the Service Bus client library for Python [API reference](#) for an example of library reference documentation that follows this guidance.

## Library index

The library index is the landing page for API reference documentation for an Azure service's client libraries. It lists the available libraries in a language for the service, provides a short description of each, and links to the overview page for each library.

- Template: [Library index template](#)
- Example: TODO

For example, the library index and library overviews for the Azure Active Directory client libraries for Python would be accessed from these nodes in the Python reference TOC:



Library index and overviews in the reference TOC

## Library overview

The library overview is the API reference landing page for a *single library* for an Azure service. It should help a developer get up and running quickly with the library. It includes package installation, client authentication, and code snippets for the most common operations developers need to perform with the library.

**TIP**

Repurpose the library's README for its library overview.

## Quickstart

A client library Quickstart helps a developer get up and running quickly *with the library*. Its primary goal is to help the developer install the library's package and perform several key coding tasks in their chosen development environment.

The intent of a library Quickstart *is not* to introduce a service and its features, though these concepts can be conveyed within the context of helping developers use the library to interact with the service. Think "code first."

For detailed instruction, see [How to write an Azure client library Quickstart](#).

# Conceptual

Conceptual documentation is typically written by content developers, PMs, and others, and describes a product and its features. This is the long-form technical content appearing as articles on docs.microsoft.com. For example, Azure's conceptual documentation resides in [azure-docs-pr](#) (and [azure-docs](#) for public contributions).

For example, conceptual documentation for an Azure service typically includes these article types:

- Overview
- Quickstarts
- Tutorials
- How-tos
- Concepts

See the [Docs Contributor Guide](#) for details on contributing conceptual content.

## Next steps

The Open Publishing System (OPS) [Onboarding & Admin Guide](#) contains extensive details about the docs.microsoft.com publishing pipeline.

### Docs tool chain

Publishing system toolchain information, including a matrix of the language-native documentation tools used in reference generation (Sphinx for Python, TypeDoc for TypeScript, etc.):

### [Supported platforms](#)

### Documenting your code

Language-specific guidance for in-code documentation (triple-slash comments, Python docstrings, etc.):

- .NET: [Preparing your .NET code for doc generation](#)
- Java: [How to Document Java APIs](#)
- JavaScript/TypeScript: [Documenting JavaScript & TypeScript APIs](#)
- Python: [How to document a Python API](#)

# How to create good docs for .NET SDK

11/2/2020 • 3 minutes to read

## Good comments make for good documents

Your .NET API comments will be transformed into public documentation on docs.microsoft.com and they'll show up in IntelliSense in the IDE. The comments should be:

- Complete – empty doc entries for methods, parameters, exceptions, and so on, make the APIs feel under-supported, temporary, or trivial.
- Correct – readers scan for critical details and become frustrated when key information is missing or incorrect.
- Contextual – readers land on this page from search and need to know how and when to use the API, and what the code implications are.
- Polished – poor or hasty grammar and spelling can confuse the reader and make even simple calls ambiguous; also, poor presentation communicates low investment.

## Best practices

1. Use `cref` for links to other types/methods.

Bad:

```
<param name="configFile"> A <a href="https://docs.microsoft.com/{path}/XmlConfigResource"></a> object...
</param> (avoid HTML)
```

Good:

```
<param name="configFile"> A <see cref="XmlConfigResource" /> object... </param>
```

2. Wrap parameter names in `<paramref>` tags.
3. If you have more than one paragraph in the doc comment, separate the paragraphs with ` `` tags).
4. Code snippets should be wrapped in `<code>` within `<example>` tags.
5. Use `<seealso>` for type/method links that have text for a "See Also" section.
6. For hyperlinks, just use the naked hyperlink:
  - for example "<https://docs.microsoft.com/azure/compute/{doc}>"

## XML Doc Tags

### List of XML Doc tags

```
<c> | <para> | <see> | <code> | <param> | <seealso> | <example> | <paramref> | <summary>
<exception> | <permission> | <typeparam> | <include> | <remarks> | <typeparamref>
<list> | <returns> | <value>
```

## Descriptions

Use these guidelines to write descriptions for the docs.

### Good method summaries/descriptions

- Start with a verb. This is an action.

- Use third person (descriptive) not second person (prescriptive) verb. The description is in third person declarative rather than second person imperative.
- Gets the label. (preferred)
- Get the label. (avoid)
- Avoid describing restrictions or other detailed aspects of the behavior of the underlying business logic if it's enforced on the server side, and not on the client.
- Difficult to maintain!
- Provide contextual information beyond regurgitating the method name:
  - Bad: `/// <summary> Sets the tool tip text. </summary>`
  - Good:
 

`/// <summary> Specifies the text to display in a tool tip. The text displays when the cursor lingers over the component. </summary>`

### Good parameter descriptions

- Keep it as short as you can (it's okay to use a phrase), but provide any essential details.
- Example:
  - Bad: `<param> The text of the tool tip. </param>`
  - Good:
 

`<param> The string to display in the tool tip. If this value is <code>null</code>, the tool tip is turned off for this component. </param>`

### Good return value descriptions

- Same as parameters – keep it reasonably short (but provide additional details), link to the class/interface that defines it (for return objects), and focus on context.
- Explicate the object returned by the call.
- If you return a collection, indicate the ordering/indexing and whether or not it was sorted (and how it was sorted).
- Any notes on the choice of collection type returned.
- Example:
  - Bad: `<returns> An array of resource configurations. </returns>`
  - Good:
 

`<returns> A <see cref="List < T >"> that contains the first 50 resource configurations, sorted in the order they were retrieved from the service. </returns>`

### Good thrown exception descriptions

- Only document-specific exceptions.
- All checked exceptions. These must be declared in the throws clause.
- Those unchecked exceptions that the caller might reasonably want to catch, including all exceptions that can be thrown, though, is just a waste of screen real-estate.)
- Provide troubleshooting or handling information when appropriate and useful.
- Examples:
  - Bad: `/// <exception cref="ConfigFileParse"> The config file could not be read. </exceptions>`
  - Good:
 

`/// <exception cref="ConfigFileParse"> The provided XML configuration file did not match the format provided by the content service schema at http://url_schema_doc. Please validate the schema before submitting it to this call.`

### Other scenarios

Documenting "internal" methods:

- `/// <summary> This method is internal and should not be used.</summary>`
  - `/// <summary> "The method is internal and only available for SDK completeness." </summary>`

```
/// <remarks> Please use <cref="ProperMethod" /> instead.</remarks>
```

- Or `<seealso>`

## Overwrites

Overwrites are hybrid YAML + Markdown files that can be used to "overwrite" any part of an autogenerated ref page, or to add Remarks and code examples. The overwrite files are in the "ref-docs-overwrites" folder in azure-docs-sdk-dotnet. Overwrite file names should be some form of the API name with overwritten content, like BatchCollection.CreateBatch\_Batch.md.

They take this format:

```
---
uid: <uid for API taken from the corresponding YAML file in the "ref-docs-autogen" directory>
<field_to_overwrite>: <string value or "*content">
---
<if *content is set, MarkDown to use for the overwrite for the indicated field is here>
```

Example:

```
---
uid: com.microsoft.applicationinsights.common._common_utils
remarks: *content
---
This API is amazingly rad, brah!
```

## OPS Docs Guidance

### Managed reference onboarding and workflow

For a detailed explanation of how to .NET reference (C# and VB) is produced, see [.NET reference requirements](#).

## See also

- [Azure ref writing style](#)

# Azure CLI article guidelines

5/5/2021 • 9 minutes to read

This article describes the style convention for writing Azure CLI articles. Following this guidance ensures that content for the Azure CLI is consistent. For more information on submitting standalone scripts and the pull request process, see:

- [Azure CLI and Azure PowerShell best practices](#)
- [Azure tools script PR review process](#)

The [Azure CLI editor's checklist](#) provides a list of *minimum requirements* for new and revised Azure CLI articles.

## Table of contents

- **name**: Omit extra words like **Azure CLI** and **Quickstart** if this information is clearly identified in the [TOC file format](#). Instead, take the opportunity to provide more information about the function of the article.
- **displayName**
  - Add this property whenever possible adding CLI reference commands. Example: `az dt` and `az maps` are used by Azure IoT but may not be found in a TOC search.
  - There isn't room for most TOC entries to say "[article name] for portal and Azure CLI". Use the **displayName** property to identify every article in a TOC that contains CLI syntax.
  - Use logical judgment as most CLI articles contain **az login** and **az group**. Only add reference commands that are unique to the article intent and Azure service.
  - Example value: `displayName: Azure CLI, az iot, az dt, az group, az account`
- **href**: Follow [hyperlink guidelines](#).
  - The Azure CLI reference and cross-service doc content are stored in a dedicated GitHub repository.
  - Adding `?view=azure-cli-latest` to Azure CLI links is not necessary.
  - Example links:  
**Good:** `/cli/azure/install-azure-cli`  
**Avoid:** `https://docs.microsoft.com/cli/azure/install-azure-cli`  
**Incorrect:** `/cli/azure/install-azure-cli ?view=azure-cli-latest`  
**Incorrect:** `https://docs.microsoft.com/en-us /cli/azure/install-azure-cli ?view=azure-cli-latest`
- **Reference node**: Add a link in the Reference node of the TOC to the Azure CLI reference content.

## Article metadata

- **ms.custom:** must contain the tag **devx-track-azurecli**
- **ms.date:** must be in the format **MM/DD/YYYY**.
  - Change the date when there is a significant or factual update.
    - Reorganizing the article
    - Fixing factual errors

- Adding new information
- Do not change the date if the update is insignificant.
- Fixing typos and formatting
- `title:` must be a unique string of 43-59 chars including spaces.
  - Do not include site identifier (it's auto-generated).
  - Use sentence case capitalizing only the first word and any proper nouns.
- `description:` should be 115-145 characters including spaces.

## Prerequisites

### Include files

- The [Use Azure Cloud Shell](#) include file is no longer recommended for Azure CLI docs. Instead, use one of three Azure CLI prerequisite include files when using three or more CLI references in an article.
  - The [CLI "Prerequisite" with H2 header](#) is best used when code blocks contain only Azure CLI syntax.
  - The [CLI "Prepare your environment" with H3 header](#) is designed to be used in conceptual tabs.
  - The CLI [include file with no header](#) works well when CLI follows other prerequisites in a bulleted list.
- Always place Azure service prerequisites first *followed by* instructions for Azure Cloud Shell and the Azure CLI. For example, "You must have `Microsoft.Authorization/roleAssignments/write` permissions to complete the instructions in this tutorial" should come before "Prepare your environment for the Azure CLI".

## Reference content

### Versions

- If the reference command requires a minimum version of the CLI higher than 2.0, mention the version in your article and sample script.
- All extension references are version-specific and often require the most recent [Azure CLI release](#).
- Double-check core references although most only require version 2.0.67 or later.

### Links

- Supply a link to the Azure CLI reference content as often as possible without being redundant.
- There should be at least one reference link for every CLI reference group within an article.
- Avoid linking to a large A-Z reference list that forces the user to scroll to find a reference command.

Here are a few examples:

	EXAMPLE	USE WHEN
Reference group:	<a href="#">az eventhubs</a>	You're writing about a command group and not referring to a single reference or command.
Reference:	<a href="#">az eventhubs namespace</a>	You're writing about a reference and not referring to a single command.
Reference command:	<a href="#">az eventhubs namespace create</a>	You're writing about a reference action.

### Content

## IMPORTANT

Do not copy reference content from the auto-generated docs found in <https://docs.microsoft.com/cli/azure/> or in <https://docs.microsoft.com/cli/azure/ext/>. Instead, provide a link to the reference article.

## Good:

"Azure CLI commands used in this article (H2)"

- [az group create](#)
- [az network vnet create](#)
- [az network vnet subnet create](#)
- [az network vnet subnet show](#)
- [az cosmosdb create](#)
- [az group delete](#)

**Incorrect:** (Notes have been copied from reference content, and although not likely to change, these descriptions will not be auto-updated.)

"Azure CLI commands used in this article (H2)"

COMMAND	NOTES
<a href="#">az group create</a>	Creates a resource group in which all resources are stored.
<a href="#">az network vnet create</a>	Creates an Azure virtual network.
<a href="#">az network vnet subnet create</a>	Creates a subnet for an Azure virtual network.
<a href="#">az network vnet subnet show</a>	Returns a subnet for an Azure virtual network.
<a href="#">az cosmosdb create</a>	Creates an Azure Cosmos DB account.
<a href="#">az group delete</a>	Deletes a resource group including all nested resources.

## Syntax formatting

Use the following short-list of applied guidelines that are designed to coincide with [PowerShell syntax elements](#):

1. Always use the full name for reference commands and parameters. Avoid using aliases unless you're specifically demonstrating the alias.
  - **Correct:** `az storage account create --name mySG --resource-group myRG`
  - **Incorrect:** `az storage account create -n mySG -g myRG`
2. Use bold for **command group names**.
  - **Correct:** This reference is part of the **baremetal-infrastructure** extension for Azure CLI and requires version 2.12.0 or higher.
3. Use backticks for `reference names` both partial and full. When writing a conceptual article (as opposed to reference content), the first instance of a reference command name should be hyperlinked to the reference documentation. Don't use backticks, bold, or other markup inside the brackets of a hyperlink.
  - **Good:** Use `az group create` to create a resource group.
  - **Better:** Use [az group create](#) to create a resource group.

- **Incorrect:** Use `az group create` to create a resource group.
4. Use backticks when showing the use of a `--parameter` within text.
- **Correct:** Use the `az group create` command specifying your desired resource group name in the `--name` parameter.
  - **Incorrect:** Use the `az group create` command specifying your desired resource group name in the `name` parameter.
  - **Incorrect:** Use the `az group create` command specifying your desired resource group name in the `name` parameter.
5. Use backticks for parameter `values`.
- **Correct:** Use `az group create --name MyResourceGroup` to create a resource group named `myResourceGroup`.
  - **Correct:** Create a resource group with `az group create`. This resource group is named `myResourceGroup` and the location is `eastus2`.
  - **Incorrect:** Use `az group create --name MyResourceGroup` to create a resource group named `myResourceGroup`.
6. Use bold when talking about a parameter by **name** referring to it as an **object**.
- **Correct:** The `az group create` command has several optional parameters including **managed-by** and **subscription**.
7. Use backticks for `file paths`, and `inline syntax examples`.
8. Use backticks for `URLs` that are not meant to be clickable in the document.
9. *Use italics for emphasis*, not for semantic markup.

## Argument format

Use the long format (`--resource-group`) instead of short format (`-g`) for all arguments.

- **Good:** `az group create --location westus --group MyResourceGroup`
- **Avoid:** `az group create -l westus -g MyResourceGroup`

## Argument order

Arguments to CLI commands should appear in the following order:

1. Resource group name (`--resource-group`)
2. Resource name or identifier (`--name`, `--id`, etc.)
3. Named/boolean arguments (no values)
4. Arguments requiring values
5. Multi-value arguments
6. Repeatable arguments
7. Tags
8. Positional arguments

Example with no positional arguments:

```
az vm create --resource-group myResourceGroup --name myVM --no-wait --image WindowsWhatever
```

Example with positional arguments:

```
az acr build --registry myregistry --image helloacrbuild:v1 ~/myappcode
```

## Variables

If possible, avoid reusing shell variables across multiple code blocks.

The reader may complete article steps in different sessions. Using variables across code blocks may cause errors (and frustration) in this situation, if they're not set correctly. If you must carry the variables across steps, make it clear that variables are reused in later steps.

## Format for Bash shell

Format your Azure CLI code blocks for the [Bash shell](#). For a good tutorial on the Bash shell, see the [Bash Beginner's Guide](#). The most common features of `bash` that you'll need to use in writing samples for the Azure CLI are:

- Line continuation character: `\`
- Variable set: `MYVAR=myvalue`
  - Don't use the `export` keyword when setting a variable. `export` can have unintended consequences.
- Variable get: `$MYVAR`
- Capture command output with `$(...)`: `MYVAR=$(execute-command)`
  - You may also capture command output with ``...`` but the `$(...)` syntax is preferred.
- Escape shell-interpreted characters with `'`: `MYVAR='literal$'`
- Force evaluation of shell-interpreted characters with `"`: `MYVAR="prefix-$OTHERVAR"`

## Randomize passwords for new resources

If you're creating a resource that will have a password associated with it, don't use a hardcoded password. Checking any passwords into source control, even examples, is a security risk.

The following methods are recommended to generate a password:

- `openssl rand --base64` will generate a series of random characters suitable for a password. Make sure the number of characters generated is divisible by 3 (use at least 15.) For example:

```
NEW_PASSWORD=$(openssl rand --base64 15)
```

- Use the `/dev/urandom` source to generate a password of arbitrary length and characters. The resulting password will meet complexity requirements. See [StackOverflow: How to generate a random string](#) for details.
- `uuidgen` will generate a new [UUID](#). Example:

```
NEW_PASSWORD=$(uuidgen)
```

If you need to discuss credentials for Linux virtual machines or scale-set instances (for example, when using `az vm create` or `az vmss create`), don't use passwords. The standard for Linux is to use an SSH key and Azure will autogenerated the SSH key if you use the `--generate-ssh-keys` parameter.

## Avoid naming conflicts

Some Azure resources, like Azure Container Registry and Key Vault, have resources tied to domain names. Those resources must have a *universally unique* name. For that reason, use a random value as part of names when uniqueness is required. If you don't, scripts will fail to create a required resource if more than one person runs them. Randomness won't *prevent* conflict, but can *mitigate* it.

Use `$RANDOM` to add a random number to a name. For example:

```
NEW_ACR_NAME="myacr-$RANDOM"
```

You can also create a random string identifier from `/dev/urandom`. See [StackOverflow: How to generate a random string](#) for details.

## Line continuation

If the command length causes horizontal scrolling, use a line continuation character and indent each following line 4 spaces. Go for readability over strict adherence to this rule. The example below uses single-line statements for two of the arguments to maintain readability.

```
az container create \
    --resource-group $RES_GROUP \
    --name acr-tasks \
    --image $ACR_NAME.azurecr.io/helloacrtasks:v1 \
    --registry-login-server $ACR_NAME.azurecr.io \
    --registry-username $(az keyvault secret show --vault-name $AKV_NAME --name $ACR_NAME-pull-usr --query
    value -o tsv) \
    --registry-password $(az keyvault secret show --vault-name $AKV_NAME --name $ACR_NAME-pull-pwd --query
    value -o tsv) \
    --dns-name-label acr-tasks-$ACR_NAME \
    --query "{FQDN:ipAddress.fqdn}" \
    --output table
```

## Output

When you show output for a command, tag the output code block as `output` (after the backticks). The output must be in a separate block from the CLI commands.

### IMPORTANT

Don't display command output in an image. Images break accessibility by defeating screen readers and interfere with web search.

```
az vm create -name MyVm -resource-group MyResourceGroup --image UbuntuLTS --generate-ssh-keys
```

An output block for the previous example command:

```
{- Finished ..
  "fqdns": "",
  "id": "/subscriptions/896e1936-a2ce-4761-9c66-5e3cec0bbba1/resourceGroups/abc/providers/Microsoft.Compute/virtualMachines/MyVm",
  "location": "eastus2",
  "macAddress": "00-0D-3A-7C-08-B0",
  "powerState": "VM running",
  "privateIpAddress": "10.0.0.4",
  "publicIpAddress": "20.4* 19.249",
  "resourceGroup": "MyResourceGroup",
  "zones": ""
}
```

## Interactive code snippets (Try It)

The use of the **azurecli-interactive** code tag is under discussion as **Try It** actually opens Azure Cloud Shell without copying the code sample. Also, code samples with [line continuation](#) characters must first be copied to a text editor, modified, and then copied back to the console. It is better to clearly explain how to install the Azure CLI in the "Prepare your environment" section of the article. If needed, add a link back to prerequisites.

## Unsupported CLI commands when using Cloud Shell

Not every Azure CLI command is supported by Azure Cloud Shell. If you mix Cloud Shell-functional commands with commands that don't work in the Cloud Shell, you risk frustrating customers when only *some* commands work. Use the **azurecli** language tag for all code blocks to help avoid confusion. For example,

```
# This command requires running the docker daemon, which is not supported in Azure Cloud Shell.
az acr login -n myregistry
```

## See also

- [Azure CLI editor's checklist](#)

# PowerShell contributor reference

11/2/2020 • 2 minutes to read

You can find detailed information on contributing PowerShell reference documentation in the [Onboarding Guide](#). The PowerShell articles explain how to contribute new content and also how to edit existing content. PowerShell content is not autogenerated, so you [use PlatyPs to generate Markdown files](#) from your modules.

Before you read more about these processes, you need to understand the differences between contributing non-Azure PowerShell content and Azure PowerShell content.

## Non-Azure PowerShell overview

To onboard new reference documentation for a non-Azure PowerShell package, you will submit an onboarding request at <https://aka.ms/publish-on-docs>. After you submit the request, the Onboarding team will get in touch with you regarding repo configuration and next steps for publication. Once your documentation goes live, your team's repo admins will control the publication process.

Read more about [Non-Azure PowerShell Documentation](#) in the Onboarding Guide, including how to use PlatyPs to prepare your documentation Markdown files.

## Azure PowerShell overview

To onboard new reference documentation for an Azure PowerShell package, do not submit an onboarding request. Instead, you will submit pull requests to the existing repo ([Azure/azure-powershell-pr](#)) and use established pipelines to publish your documentation. New Azure PowerShell documentation is built once every two weeks on a set schedule. Pull request updates to existing documentation are merged to master and live once a day, Monday through Friday.

See the Onboarding Guide for instructions on how to onboard and edit [Azure PowerShell documentation](#). Learn more about the [OpenAPI Hub](#) specifications and pipeline in the [Azure GitHub wiki](#).

To learn how to write more consistent samples and scripts, see [Azure PowerShell code conventions](#).

# Azure client library index

4/16/2021 • 2 minutes to read

The library index is the landing page for an Azure service's library reference on docs.microsoft.com. It lists and describes the libraries available for the service and links to the library reference landing pages. See [Document your library - library index](#) for more information.

**Title:** The H1 of your library index should be in the format: # [Service Name] client libraries for [Language]

- Example: # Azure Batch client libraries for Python

**Introduction:** The introduction appears directly under the title (H1) of the index.

- **DO NOT** use an "Introduction" or "Overview" heading (H2) for this section.
- Describe the service briefly. You can usually use the first line of the service's docs landing page for this (Example: [Cosmos DB docs landing page](#)).

Present a table in the *Libraries* section listing each library and their packages available for the service. If there are multiple versions of your library in common use, provide a table for each version available. Use an H3 header above each table that includes the version number. To help the user understand whether the package they are using is up-to-date, include the version number even if only one version is supported.

## TIP

A "library" is generally the intersection of a service and language. Many libraries publish multiple "packages".

Use this Markdown starter for your table:

*Note the samples column can be skipped if there are no dedicated samples for the packages like in the case of libraries for resource management*

Nuget Package	Reference	Samples
[Package Name](PATH/TO/PACKAGE/REF/LANDING-PAGE)	[API Reference](https://link-to-package-manager)	[Samples](https://github.com/azure/link-to-source-code/samples)

## Libraries for resource management

Use the following library to work with the Azure Storage resource provider:

NUGET PACKAGE	REFERENCE
<a href="#">Microsoft.Azure.Management.Storage</a>	<a href="#">API Reference for Microsoft.Azure.Management.Storage</a>

## Libraries for data access

For example, here's the library index for Storage:

## Version 12.x

The version 12.x client library for .NET is part of the Azure SDK for .NET. The source code for the Azure Storage client libraries for .NET is available on [GitHub](#).

Use the following version 12.x packages to work with blobs, files, and queues:

NUGET PACKAGE	REFERENCE	SAMPLES
Azure.Storage.Blobs.Batch		Samples for Azure.Storage.Blobs.Batch
Azure.Storage.Blobs.Cryptography		Samples for Azure.Storage.Blobs.Cryptography
Azure.Storage.Blobs	API Reference for Azure.Storage.Blobs	Samples for Azure.Storage.Blobs
Azure.Storage.Common		Samples for Azure.Storage.Common
Azure.Storage.Files.DataLake	API Reference for Azure.Storage.Files.DataLake	Samples for Azure.Storage.Files.DataLake
Azure.Storage.FilesShares	API Reference for Azure.Storage.FilesShares	Samples for Azure.Storage.FilesShares
Azure.Storage.Queues	API Reference for Azure.Storage.Queues	Samples for Azure.Storage.Queues

## Version 11.x

The source code for the Azure Storage client library for .NET is available on [GitHub](#).

Use the following version 11.x packages to work with blobs, files, and queues:

NUGET PACKAGE	REFERENCE	SAMPLES
Microsoft.Azure.Storage.Blob	API Reference for Microsoft.Azure.Storage.Blob	Samples for Microsoft.Azure.Storage.Blob
Microsoft.Azure.Storage.Common	API Reference for Microsoft.Azure.Storage.Common	Samples for Microsoft.Azure.Storage.Common
Microsoft.Azure.Storage.File	API Reference for Microsoft.Azure.Storage.File	Samples for Microsoft.Azure.Storage.File
Microsoft.Azure.Storage.Queue	API Reference for Microsoft.Azure.Storage.Queue	Samples for Microsoft.Azure.Storage.Queue

# Azure reference writing style

4/27/2021 • 2 minutes to read

## Why write good API comments

Your API comments will be transformed into public documentation on docs.microsoft.com. This means that your comments should be:

- Complete – empty doc entries for methods, parameters et al make the APIs feel undersupported, temporary, or trivial.
- Accurate – readers scan for critical details and become frustrated when key information is not present or incorrect (see: missing default values, ranges, et al).
- Contextual – readers land on this page from search and need to know how and when to use the API, and what the code implications are.
- Polished – poor or hasty grammar and spelling can confuse the reader and make even simple calls ambiguous; also, poor presentation communicates low investment.

## The 9 essentials for baseline Azure API Ref comments

1. Always start a sentence (or fragment) with a capital letter and end it with a period.
2. Document every API completely.
3. Always try to provide enough summary details to make the specific use of the API clear when scanning.
  - Corollary: Don't just repeat the API name in sentence form. Add a second (or more!) sentence providing context!
4. Provide any information that is not immediately apparent from the syntax and would require knowledge of the client (not server) code to successfully call the API. Simply, be thorough and concise.
5. All abbreviations should be in all-capital letters, unless you are referring to a symbol.
  - Examples:
    - "API, not "Api".
    - "ID", not "Id".
6. Avoid arbitrary capitalization. Only capitalize proper nouns.
  - Examples:
    - "API Management service", not "Api management service" or "API Management Service".
    - "SKU" not "Sku", "IoT" not "iot" or "IOT", "URL" not "Url".
7. Reasonable brevity (but not opaque brevity) is good. Remove unnecessary words.
  - Article: [The 25-Word Limit](#)
8. Run a grammar checking tool for your individual comments before you commit the changes. Fix any obvious grammar and spelling errors.
9. Use the Oxford comma (commas used to separate all elements in a list). It's better for technical documentation. For example: "...contains the ID, the token, and a display name", not "...contains the ID, the token and a display name".

# Unified Content Model

5/19/2021 • 19 minutes to read

The unified content model is an effort to ensure the content on Docs.microsoft.com is structured in a way that supports the business and customers. Structured content is more sustainable, easier to reuse, more efficient to manage, and supports sophisticated technical and customer experiences. The unified content model will make it easier for customers to understand and use our content, will make it easier for us to deploy and manage content, and will make it easier for us to build innovative experiences.

This guide details all of the current content types. Contact [devrelia@service.microsoft.com](mailto:devrelia@service.microsoft.com) if you think something on this page should change.

## NOTE

We have removed tentative content types in this version of the documentation, because including them was causing confusion. This version contains content types that have already been designed in partnership with the Product team and engineering. If a content type is not represented here, it still may exist on the site currently, but it isn't structured and isn't yet planned for in the new content model.

## Achievement

Status: Live



B A D G E

### Get started with Power Automate

Completed on 1/15/2019

A recognition awarded to a User who has completed a gamification milestone.

#### Live Example

*No current example*

#### When to use Achievements

You want to incentivize users to hit a milestone or you want to promote some other gamified behavior.

#### Elements of Achievements

- Title
- Description
- Icon

### Taxonomies used on Achievements

- Achievement Type

### Relationships with other content types

- Achievement this learning path awards
- Achievements this user has earned
- Achievement this module awards

## Answer

Status: Development

An answer a user or a support rep has provided.

### Live Example

*Q&A is being redesigned. New answers are not yet live on the site.*

### When to use Answers

You need to an answer a user has provided on the Q&A platform. Present answers with the question they're answering for context. So that users can get useful answers to their questions.

### Elements of Answers

- Attached content
- Body
- Comment count
- Date
- URL
- Prior versions of this answer
- Most recent version date
- Count of people who found this answer helpful

### Taxonomies used on Answers

- none

### Relationships with other content types

- Answers this user has provided
- User who provided this answer
- Answers for this question
- Answer this comment is clarifying
- Parent question for this answer
- Comments on this answer

## Architecture

Status: Live

Intelligent apps using Azure Database for MySQL

Filter by title

- Intelligent apps using Azure MySQL
- Intelligent apps using Azure PostgreSQL
- Interactive querying with HDInsight
- Loan charge-off prediction with HDInsight Spark
- Loan charge-off prediction with SQL Server
- Loan credit risk modeling
- Loan credit risk with SQL Server
- Master data management with CluedIn
- Master data management with Profisee
- Messaging
- Modern data warehouse
- N-tier app with Cassandra
- Ops automation using Event Grid
- Oracle migration to Azure
- Personalization using Cosmos DB
- Retail and e-commerce using Azure MySQL
- Retail and e-commerce using Azure PostgreSQL
- Retail and e-commerce using Cosmos DB

[Download PDF](#)

[Download an SVG of this architecture.](#)

```

graph LR
    EH[Event Hubs] -- 1 --> FA[Function Apps]
    SA[Storage Account] -- 1 --> FA
    FA -- 2 --> ACS[Azure Cognitive Services Text Analytics API]
    ACS -- 3 --> ADMS[Azure Database MySQL Server]
    ADMS -- 4 --> AMLS[Azure Machine Learning Studio]
    AMLS -- 5 --> PB[Power BI]
  
```

Intelligent apps using Azure Database for MySQL

App Service | Cognitive Services | Database for MySQL | Machine Learning | Power BI

Solution Idea

If you'd like to see us expand this article with more information, implementation details, pricing guidance, or code examples, let us know with GitHub Feedback!

Develop sophisticated, transformational apps using state of the art machine learning algorithms and integrated visualization tools to get actionable insights and analytics.

## Architecture

Is this page helpful?

Yes No

Planning guidance for designing and deploying an end-to-end solution for a particular business problem, scenario, or workload.

## Live Example

[Live Architecture](#)

## When to use Architectures

You want to give customers an example or recommended architecture for a technical solution.

## Elements of Architectures

- Body
- Description
- Title

## Taxonomies used on Architectures

- Product
- Azure Category

## Relationships with other content types

- Architectures in this collection

## Article

Status: Planned

Content that explains a concept or idea. All existing documentation content is considered an article by default.

## Live Example

[Live Article](#)

## When to use Articles

You have documentation content with little structure to it.

## Elements of Articles

- Attached content

- Body
- Title

#### Taxonomies used on Articles

- Product
- Dev Language
- Role

#### Relationships with other content types

- Articles to address issues raised in this assessment
- Articles in this collection

## Assessment Session

Status: Development

The instance of a self-assessment or a renewal assessment triggered by a user starting a self-assessment or renewal assessment.

#### Live Example

*No current example*

#### When to use Assessment Sessions

You are using any assessment interaction on Docs. These assessments should be handled by a questionnaire, start an assessment session, and result in a report. Choose this so that a User can have a 1:1 relationship with an assessment session and its corresponding report. A user may take several kinds of self-assessments, or may take a single self-assessment many times, providing different answers and getting different reports.

#### Elements of Assessment Sessions

- Created date
- Date
- Name
- Completion date

#### Taxonomies used on Assessment Sessions

- none

#### Relationships with other content types

- Questionnaires used in this assessment session
- Assessment session that generated this report
- User who initiated this assessment session
- Completed assessment sessions for this self-assessment and this user
- Assessment sessions this user has in progress
- In-progress assessment sessions for this self-assessment and this user

## Certification

Status: Live

The screenshot shows a Microsoft certification page for the 'Power Platform App Maker Associate' exam. At the top, there's a 'Microsoft CERTIFIED ASSOCIATE' logo with two stars. Below it, the title 'Microsoft Certified: Power Platform App Maker Associate' is displayed. A note states: 'In response to the coronavirus (COVID-19) situation, Microsoft is implementing several temporary changes to our training and certification program. Learn more.' A description follows: 'The app maker builds solutions to simplify, automate, and transform tasks and processes for themselves and their team where they have deep expertise in the solution domain. They are skilled in key technical business analyst tasks such as data modeling, basic UX design, requirements analysis, and process analysis.' Below this, a note says: 'Note: This certification will be available around July 2020.' It lists job roles: Business Analyst, Business Owner, Business User, Data Analyst, Developer, DevOps Engineer, Functional Consultant, App Maker, Technology Manager. Required exams are listed as PL-100. An 'Important' section links to details, and a 'Go to Certification Dashboard' button is present. The main section is titled 'Certification details' and contains two boxes: 'Take one exam' (with a 'CERTIFICATION EXAM Microsoft Power Platform App Maker (beta)' icon) and 'Earn the certification' (with an 'ASSOCIATE CERTIFICATION Microsoft Certified: Power Platform App Maker Associate' icon). Below these are sections for 'Skills measured' (listing: Design solutions, Create solutions, Analyze data, Implement and manage solutions) and a link to 'Download certification skills outline'.

A Microsoft-branded offering that confirms that a user has mastered a set of concepts related to a topic.

## Live Example

### Live Certification

### When to use Certifications

You are representing official Microsoft certifications.

### Elements of Certifications

- Pre-requisites
- Optional Pre-requisites

### Taxonomies used on Certifications

- Product
- Role
- Level
- Certification Type

### Relationships with other content types

- Certification this exam counts toward
- Exam(s) required for this certification
- Certification this exam counts toward
- Certifications in this collection
- Certification this user certification corresponds to
- Exam(s) required for this certification

## Challenge

Status: Live

## Microsoft Ignite App Maker Challenge

Do you want to create apps to help make your business more efficient? Then this path is for you. It introduces you to Power Apps, helps you create & customize an app, then manage and distribute it. Complete one challenge in time and you will get notified on October 15th about your free Microsoft Certification exam.



This challenge ended on October 7, 2020

[Start learning](#)

### Final leaderboard

31,540 participants

Modules completed

	9/9
-- 68歲懶表姨婆	9/9
# #Fight For Justice @ @HK	9/9
Necdet Saritas	9/9

A time-based competition that a set of users elects to participate in. Each challenge can evaluate users' completion of a certain collection of Modules and/or Learning Paths *or* the total XP that users earn during the duration of the Challenge.

### Live Example

### Live Challenge

### When to use Challenges

You want to create an interactive event for a group of end users.

### Elements of Challenges

- Description
- End Date
- Leaderboard
- Start Date
- Title
- Type
- Teams

### Taxonomies used on Challenges

- Solution

### Relationships with other content types

- Users participating in this challenge
- Challenge that uses this collection
- Challenges this user has participated in
- Collection that must be completed for this challenge
- Challenges this user is currently participating in

## Channel

Status: Development

A place on the site with a sequence of content being broadcast.

### Live Example

No current example

## When to use Channels

To give standard places to see video content.

## Elements of Channels

- Name

## Taxonomies used on Channels

- none

## Relationships with other content types

- Event being streamed on this channel
- Channel(s) this event is being livestreamed on
- Channels in this collection
- Collections this channel is in

## Code Sample

Status: Live

The screenshot shows a Microsoft Docs page for a code sample titled ".NET Core WinForms Formatting Utility (C#)". The page is dated 05/15/2020. It features a "Browse code" button and a "Download ZIP" button. The content includes a brief description of the application, which is a .NET Core Windows Forms application written in C# for applying numeric or date/time format strings. Below this, there are sections for "Sample prerequisites" (mentioning .NET Core 3.0 SDK), "Building the sample" (instructions for downloading and building the project), and a note about the source code being in MSBuild format (.csproj).

Deployable code presenting a usable technical solution. Always associated with Source Code.

## Live Example

[Live Code Sample](#)

## When to use Code Samples

You want to give users explanatory information about source code and provide downloadable source code.

## Elements of Code Samples

- Description
- Title

## Taxonomies used on Code Samples

- Dev Language

- Product
- Platform

## Relationships with other content types

- Code samples in this collection
- Source code for this code sample

# Collection

Status: Live

The screenshot shows a Microsoft Collection page titled "Example". The page header includes the Microsoft logo, navigation links for Docs, Documentation, Learn (which is underlined), Q&A, and Code Samples, and a search bar. Below the header, the breadcrumb navigation shows "Docs / My profile / Collections / Example". The main content area features a title "Example" with a subtitle "39 min • 1 Module • Created by Sarah Barrett". A progress bar indicates "100% completed". Below this, a section titled "Items in this collection" displays a single item: "Create an Azure account" (Module, 39 min, 4.7 stars, 93,459 reviews). Another progress bar for this item is also at 100% completed. At the bottom of the page, there's a link "Browse to add more" and a footer with language selection ("English (United States)"), theme options, and links to Privacy & Cookies, Terms of Use, Site Feedback, Trademarks, and © Microsoft 2020.

A set of content items that is organized around a set of unifying criteria. These criteria are editorially driven, not based on metadata, and individual items have to be added to each collection.

### Live Example

[Live Collection](#)

### When to use Collections

You want to assemble a set of content based on manual curation and show it to users, reuse it elsewhere on the site.

### Elements of Collections

- Created date
- Description
- External Link
- Is Microsoft?
- Last modified date
- Number of items in this collection
- Title
- Custom branding

### Taxonomies used on Collections

- Dev Language

- Level
- Product
- Role
- ms.prod/technology
- ms.service/subservice
- salesplay
- ms.topic
- ms.devlang

### **Relationships with other content types**

- Collections to prepare for this exam
- Challenge that uses this collection
- Code samples in this collection
- Architectures in this collection
- User who created this collection
- Collection of content for this episode
- Lifecycles in this collection
- Channels in this collection
- Collection used to study for this renewal assessment
- Reference articles in this collection
- Articles in this collection
- Learning paths in this collection
- Pre-read collection for this self-assessment
- Self-assessments in this collection
- Collection of content for this episode
- Modules in this collection
- FAQs in this collection
- Collections this user has created
- Exam this collection supports
- Collections this channel is in
- Courses in this collection
- Questions in this collection
- Certifications in this collection
- Collection that must be completed for this challenge

## **Comment**

**Status:** Development

A clarification another user asks for or provides on a question or answer.

### **Live Example**

*No current example*

### **When to use Comments**

To make questions easier to answer and answers more helpful.

### **Elements of Comments**

- Body
- Date

- Most recent version date
- Upvote count

## Taxonomies used on Comments

- none

## Relationships with other content types

- Comments on this question
- Answer this comment is clarifying
- Comments this user has provided
- Comments on this answer
- User who provided this comment
- Question this comment is clarifying

# Course

Status: Live

The screenshot shows a web browser window displaying a Microsoft course page. The title of the course is "Course AZ-400T00-A: Designing and Implementing Microsoft DevOps solutions". Below the title, it says "5 Days • Instructor-led training • Advanced • English". A description of the course follows, stating: "This course provides the knowledge and skills to design and implement DevOps processes and practices. Students will learn how to plan for DevOps, use source control, scale Git for an enterprise, consolidate artifacts, design a dependency management strategy, manage secrets, implement continuous integration, implement a container build strategy, design a release strategy, set up a release management workflow, implement a deployment pattern, and optimize feedback mechanisms." Below the description, there is a section titled "Audience profile" which states: "Students in this course are interested in implementing DevOps processes or in passing the Microsoft Azure DevOps Solutions certification exam." It also lists the "Job role: DevOps Engineer", "Preparation for exam: AZ-400", and "Features: none". A blue button labeled "Find a learning partner >" is visible. At the bottom, there is a section titled "Skills gained" with a list of three items: "Plan for the transformation with shared goals and timelines", "Select a project and identify project metrics and KPIs", and "Create a team and agile organization structure". A link "Show more ▾" is located below this section.

A partner-provided offering that teaches people the key concepts and topics that will be used in an Exam in order to earn a Certification. Courses may be in-person or online.

## Live Example

[Live Course](#)

## When to use Courses

You're representing an official Microsoft course that supports an exam, leading to a certification. Other kinds of online training are not courses.

## Elements of Courses

- Description
- Duration
- ID
- Outline

- Pre-requisites
- Skills
- Title
- Audience Profile

#### **Taxonomies used on Courses**

- Role
- Level
- Language

#### **Relationships with other content types**

- Courses to prepare for this exam
- Ways to take this course
- Exams this course prepares you for
- Content this course covers
- Courses in this collection

## Course Offering

Status: Development

A scheduled instance of a course provided by a partner. A course offering happens at a scheduled date/time and may happen in-person or online.

#### **Live Example**

*No current example*

#### **When to use Course Offerings**

You are representing an official instance of a course provided by a partner. Other kinds of in-person training aren't covered by this content type yet.

#### **Elements of Course Offerings**

- Date
- ID
- Location (City, State)
- Price
- Title
- Session Duration
- Meeting Frequency
- Start Time
- Start Date
- End Date

#### **Taxonomies used on Course Offerings**

- none

#### **Relationships with other content types**

- Ways to take this course
- Content this course covers

## Episode

Status: Development

An informative video within a larger, ongoing show.

### **Live Example**

*No current example*

### **When to use Episodes**

You need to present a video with useful context around it, as part of a Show.

### **Elements of Episodes**

- Date
- Description
- Duration
- Presenters
- Title
- Video
- Rating

### **Taxonomies used on Episodes**

- Dev Language
- Level
- Product
- Role

### **Relationships with other content types**

- Upcoming episodes this presenter will host
- Collection of content for this episode
- Episodes this presenter hosted
- Presenters in this episode
- Episodes in this user's queue
- Episodes in this user's history
- Show this episode is a part of
- Episodes of this show

## **Event**

**Status:** Development

A time-based event that customers can register for and attend.

### **Live Example**

*No current example*

### **When to use Events**

You want to allow people to anticipate, attend, and find content from Microsoft events.

### **Elements of Events**

- End Date
- Location (City, State)
- Name
- Start Date
- Year
- Event Brand

## Taxonomies used on Events

- none

## Relationships with other content types

- Event being streamed on this channel
- Event this session is part of
- Channel(s) this event is being livestreamed on
- Users registered for this event
- Sessions that make up this event

## Exam

Status: Live

The screenshot shows a web browser window titled "Exam PL-100: Microsoft Power Platform App Maker (beta)". The page features a blue shield logo with "Microsoft" and "EXAM" on it. The main title is "Exam PL-100: Microsoft Power Platform App Maker (beta)". A note at the top states: "In response to the coronavirus (COVID-19) situation, Microsoft is implementing several temporary changes to our training and certification program. [Learn more.](#)" Below this, there are four sections of text describing the exam's purpose and requirements. At the bottom, there is a "Note: This exam will be available around July 2020." followed by links for requirements, related exams, and important details. A "Schedule exam" button is visible at the bottom left, and a language dropdown menu at the bottom right set to "United States".

A test created by Microsoft that is connected to specific Certification. The exam evaluates a user's understanding of a specific set of concepts/topics.

## Live Example

[Live Exam](#)

## When to use Exams

You are representing an official Microsoft Exam that awards a Certification. Other kinds of online tests are not Exams.

## Elements of Exams

- Title
- Description
- Number
- Skills
- Retirement Date
- Resources

## Taxonomies used on Exams

- Product
- Role
- Level
- Language

## Relationships with other content types

- Collections to prepare for this exam
- Certification this exam counts toward
- Courses to prepare for this exam
- Learning paths to prepare for this exam
- Exams this course prepares you for
- Exam(s) required for this certification
- Certification this exam counts toward
- Exam this collection supports
- Exam(s) required for this certification

## FAQ

### Status: Live

**FAQ — Upgrading from Skype for Business to Microsoft Teams**

02/04/2021 • Applies to: Microsoft Teams

**Is there a firm deadline by which customers need to move from Skype for Business Online to Teams?**

Yes. [Skype for Business Online will be retired](#) on July 31, 2021, at which point it will no longer be accessible or supported. We encourage Skype for Business Online customers to start using Teams and begin planning their upgrades now to allow ample time to complete upgrade prior to the retirement date.

**Is Skype Meeting Broadcast going to retire at the same time as Skype for Business online?**

Yes. [Teams Live Events](#) is the successor solution to Skype Meeting Broadcast.

**How long will it take to upgrade my organization to Teams?**

Your organization's journey from Skype for Business to Teams can be defined by you. To assist in your planning and execution, we've developed comprehensive upgrade guidance based upon a proven framework designed to help you navigate the technical and organizational elements of change. Start your journey by familiarizing yourself with our [upgrade success framework](#) and associated resources that serve as the cornerstone for navigating your journey from Skype for Business to Teams.

**Is there a recommended upgrade path for Skype for**

An authored page that presents content in a question and answer format. The FAQ structure ensures optimal presentation in search results.

## Live Example

### Live FAQ

#### When to use FAQs

You want to present a list of questions, each with a single answer. Do not use an FAQ for questions that have multiple answers.

#### Elements of FAQs

- Description
- Question

- Title
- Answer

#### Taxonomies used on FAQs

- Product

#### Relationships with other content types

- FAQs in this collection

## Group

Status: Development

A set of users with certain permissions associated with them.

#### Live Example

*No current example*

#### When to use Groups

So admins can give other users moderator or admin powers.

#### Elements of Groups

- Description
- Name
- Permissions

#### Taxonomies used on Groups

- none

#### Relationships with other content types

- Users in this group
- Group this user is in

## Learning Path

Status: Live

The screenshot shows a web browser window displaying the Microsoft Learn page for the 'Evolve your DevOps practices' learning path. The page has a light gray header with the title 'Evolve your DevOps practices' and a blue circular icon with a gear and gear icon. Below the header, there's a brief description of DevOps and its benefits. A list of goals for the learning path is provided, followed by a note about preparing for the 'Exam AZ-400: Microsoft Azure DevOps Solutions'. There are tabs for 'Intermediate', 'DevOps Engineer', 'Administrator', 'Developer', and 'Azure DevOps'. A 'Start' button and a 'Bookmark' link are at the bottom. A sidebar on the left lists 'Modules in this learning path' with a single module titled 'Assess your existing software development process'.

Structured self-paced learning content with learning goals and tracked progress. Small collections of modules (3-8) that are presented in a recommended order. Users do not need to do them in order, and we do not discourage or punish skipping around.

## Live Example

### [Live Learning Path](#)

## When to use Learning Paths

You need to group together a small set of modules for a specific, official, learning objective. If you need to create an informal, ad-hoc grouping of modules, use a Collection.

## Elements of Learning Paths

- Attached content
- Card Description
- Pre-requisites
- Summary
- Title
- Video

## Taxonomies used on Learning Paths

- Product
- Role
- Level

## Relationships with other content types

- Achievement this learning path awards
- Learning paths to prepare for this exam
- Modules on this learning path
- Learning paths in this collection
- Learning paths this module is on

# Lifecycle

Status: Live

## Dynamics 365 Project Operations

Dynamics 365 Project Operations follows the [Modern](#) Lifecycle Policy.

### Support Dates

Listing	Start Date	Retirement Date
Dynamics 365 Project Operations	10/01/2020	In Support

Information on the support trajectory of a Microsoft offering.

### Live Example

*No current example*

### When to use Lifecycles

You're documenting official information about support for Microsoft offerings.

### Elements of Lifecycles

- Abstract
- Description
- Title
- Family
- Group
- Editions
- StartDate
- EndDate

### Taxonomies used on Lifecycles

- Product
- Policy Type

### Relationships with other content types

- Lifecycles in this collection

# Module

Status: Live

The screenshot shows a Microsoft Learn module titled "Manage virtual machines with the Azure CLI". The module has a rating of 4.7 (4,063 reviews) and 1000 XP. It includes sections for prerequisites, learning paths, and three exercises. The exercises are: "What is the Azure CLI?", "Exercise - Create a virtual machine", and "Exercise - Test your new virtual machine".

Modules are the building blocks of the Microsoft Learn experience containing a collection of related Units, that teach a concept using textual content, videos, and labs. They are reusable and searchable. They can be shared across multiple learning paths. They do not need to be on a learning path.

## Live Example

### [Live Module](#)

## When to use Modules

You want to teach a relaxed learner in a series of clear steps, with knowledge checks and/or interactivity. Troubleshooting content, or other information that is likely to be of most use in a high-stress situation, should not be housed in a Module.

## Elements of Modules

- Abstract
- Attached content
- Card Description
- Pre-requisites
- Summary
- Title
- Video
- XP

## Taxonomies used on Modules

- Product
- Role
- Level

## Relationships with other content types

- Units in this module
- Modules on this learning path
- Content for this report
- Modules in this collection

- Learning paths this module is on
- Achievement this module awards
- Modules recommended for this questionnaire
- Modules this user has in progress
- Module this unit is in

## Presenter

**Status:** Development

A person who publicly hosts content on Microsoft's behalf.

A small number of people probably need a page to promote them (including activity linked from a user profile.) Most presenters will be metadata where you can find other episodes and sessions they participated in.

### Live Example

*No current example*

### When to use Presenters

You need to represent people who represent Microsoft, to allow users to identify and follow interesting people

### Elements of Presenters

- Name
- Bio
- Twitch link
- Twitter link
- Youtube link

### Taxonomies used on Presenters

- none

### Relationships with other content types

- Upcoming episodes this presenter will host
- Shows this presenter hosts
- Sessions this presenter gave
- Episodes this presenter hosted
- Presenters in this episode
- Upcoming sessions from this presenter
- Presenters of this session
- Presenters of this show

## Question

**Status:** Development

A question a user wants to ask of support or other users.

### Live Example

*Q&A is being redesigned. New Questions are not yet live on the site.*

### When to use Questions

You need to display questions submitted by end users. Questions should not be used as a publishing channel.

### Elements of Questions

- Answer count

- Answered (boolean)
- Attached content
- Body
- Comment count
- Date
- Follower count
- Internal tags for this question
- Most recent version date
- Prior versions of this question
- Title
- Upvote count
- URL
- Tags for this question

### **Taxonomies used on Questions**

- Product

### **Relationships with other content types**

- Other questions related to this tag
- Questions this user has asked
- Answers for this question
- Comments on this question
- Questions using this tag
- Recommended tags for your question
- Parent question for this answer
- Tags for this question
- Questions in this collection
- User who asked this question
- Question this comment is clarifying

## **Questionnaire**

**Status:** Development

One or more questions in an assessment for a user to answer. It may have additional information in the form of text, or a video. May be oriented around a Learning Objective.

### **Live Example**

*No current example*

### **When to use Questionnaires**

You are writing an assessment. Assessments don't accept any other content types and Assessment Questions can't be used anywhere else.

### **Elements of Questionnaires**

- Answer
- Assessment Question
- Body
- Description
- Name
- Note

- Video
- Manual ID
- Recommendation text
- Recommendation link
- Publication status

#### Taxonomies used on Questionnaires

- none

#### Relationships with other content types

- Questionnaires used in this assessment session
- Questionnaires for this Renewal Assessment
- Questionnaires for this self-assessment
- Modules recommended for this questionnaire

## Renewal Assessment

#### Status: Live

An online test a user can take to extend their existing certification by one year. Users only see the tests they're eligible for.

#### Live Example

*No current example*

#### When to use Renewal Assessments

Users need to extend their official Microsoft certification online and for free.

#### Elements of Renewal Assessments

- Description
- Title

#### Taxonomies used on Renewal Assessments

- none

#### Relationships with other content types

- Reports for this renewal assessment
- Questionnaires for this Renewal Assessment
- Collection used to study for this renewal assessment
- Renewal assessments this user has completed
- Renewal assessment that would extend this user certification

## Report

#### Status: Development

The information that is automatically generated by a user completing an Assessment Session. The precise nature of the report should be defined by the self-assessment or recertification exam it's associated with.

#### Live Example

*No current example*

#### When to use Reports

You are using any assessment interaction on Docs. These should be handled by a questionnaire, start an assessment session, and result in a report.

## Elements of Reports

- Overall result
- Questionnaire result(s)
- Questionnaire recommendations
- Top recommendations

## Taxonomies used on Reports

- none

## Relationships with other content types

- Reports for this renewal assessment
- Assessment session that generated this report
- Content for this report
- Reports for this self-assessment
- Reports this user has generated

# Self-Assessment

## Status: Live

The screenshot shows the Azure Well-Architected Review interface. At the top, there's a navigation bar with Microsoft, Docs, Documentation, Learn, Q&A, and Code Samples. Below that is a search bar and a sign-in link. The main area has tabs for Assessments (PREVIEW), Available assessments, FAQ & Help, and a 'Save' and 'Share' button. The page title is 'Azure Well-Architected Review' with a subtitle 'Azure Well-Architected Review - May 18, 2021 - 12:47:11 PM'. It shows a progress bar indicating 0 of 0 questions. A note says 'Before you get started, consider Signing in to save your progress.' The main content area is titled 'Azure Well-Architected Review' and describes examining workload through lenses of reliability, cost management, operational excellence, security, and performance efficiency (30 minutes). It asks for an 'Assessment name' (set to 'Azure Well-Architected Review - May 18, 2021 - 12:47:11 PM'). Below that, it says 'Choose your interests' and lists five categories with checkboxes:

- Cost Optimization: An effective architecture achieves business goals and ROI requirements while keeping costs within the allocated budget.
- Operational Excellence: To ensure that your application is running effectively over time, consider multiple perspectives, from both an application and infrastructure angles. Your strategy must include the processes that you implement so that your users are getting the right experience.
- Performance Efficiency: Prioritize scalability as you design and implement phases. Scalability leads to lower maintenance costs, better user experience, and higher agility.
- Reliability: In a cloud environment you scale out rather than buying higher-end hardware to scale up. While it's always desirable to prevent all failure, focus your efforts in minimizing the effects of a single failing component.
- Security: Security is one of the most important aspects of any architecture. It provides confidentiality, integrity, and availability assurances against deliberate attacks and abuse of your valuable data and systems. Losing these assurances can negatively impact your business operations and revenue, as well as your organization's reputation in the marketplace.

A 'Next →' button is at the bottom right.

An interactive wizard that provides content recommendations based on user input provided via one or more questionnaires.

## Live Example

### Live Self-Assessment

#### When to use Self-Assessments

You want users to answer a set of questions in exchange for content recommendations.

#### Elements of Self-Assessments

- Description

- Name

#### Taxonomies used on Self-Assessments

- Product

#### Relationships with other content types

- Articles to address issues raised in this assessment
- Completed assessment sessions for this self-assessment and this user
- Questionnaires for this self-assessment
- Pre-read collection for this self-assessment
- Self-assessments in this collection
- In-progress assessment sessions for this self-assessment and this user
- Reports for this self-assessment
- Self-assessments this user has completed

## Session

Status: Development

A specific piece of content presented at an event.

#### Live Example

*No current example*

#### When to use Sessions

You want to convey useful information to event attendees.

#### Elements of Sessions

- Description
- Duration
- Presenters
- Start Time
- Title
- Type
- Video
- Day of event
- End time
- Session ID

#### Taxonomies used on Sessions

- Dev Language
- Level
- Product
- Role

#### Relationships with other content types

- Event this session is part of
- Sessions this presenter gave
- Sessions that make up this event
- Upcoming sessions from this presenter
- Presenters of this session
- Collection of content for this episode

- Sessions in this user's history

## Show

Status: Development

An ongoing set of video content organized around a certain theme.

### Live Example

*No current example*

### When to use Shows

You are grouping similar video content together and establishing a brand over time. You are putting together a series of videos about a particular topic, or with a particular person. These should be analogous to TV shows. One video cannot be part of multiple shows.

### Elements of Shows

- Description
- Image
- Title
- Episode Count
- Last episode date
- Locale
- Presenters
- Sort order
- Syndication feed
- Learn TV schedule

### Taxonomies used on Shows

- Product
- Dev Language
- Role
- Level
- Subject

### Relationships with other content types

- Shows this presenter hosts
- Users who subscribe to this show
- Shows this user is subscribed to
- Show this episode is a part of
- Presenters of this show
- Episodes of this show

## Tag

Status: Development

A term used to group questions.

### Live Example

*No current example*

### When to use Tags

So answerers can find questions to an answer based on tag.

## Elements of Tags

- Description
- Image
- Display label
- Slug
- Synonyms
- Common misspellings
- Sibling tags
- Parent tags
- Child tags
- Question count
- Visible (boolean)

## Taxonomies used on Tags

- none

## Relationships with other content types

- Other questions related to this tag
- Tags this user is following
- Questions using this tag
- Recommended tags for your question
- Tags for this question

## Unit

Status: Live

The screenshot shows a web browser window displaying a Microsoft Learn module. The title bar reads "Meet the team - Learn | Microsoft". The URL in the address bar is <https://docs.microsoft.com/en-us/learn/modules/assess-your-development-process/2-meet-the-team>. The page header includes "Docs / Learn / Browse / Evolve your DevOps practices / Assess your existing software development process / Meet the team". On the right, there are "Bookmark" and "LEVEL 6" buttons, and a progress bar showing "900/11299 XP". The main content area has a title "Meet the team" with a "4 minutes" duration indicator. Below the title is a paragraph about DevOps features and a team introduction. A section about Tailspin Toys follows, mentioning their new racing game and space shooter. Another section discusses the team's website build process. At the bottom is a graphic for "SPACE GAME" with the subtitle "An example site for learning".

A “chapter” of a Module. Each Unit contains a chunk of information that supports the Module. Units are not reusable and are set to NOINDEX, NOFOLLOW.

## Live Example

[Live Unit](#)

## When to use Units

You are putting together a module. Every module is composed of individual units.

## Elements of Units

- *Elements TBD*

## Taxonomies used on Units

- none

## Relationships with other content types

- Units in this module
- Module this unit is in

# User

Status: Live

The screenshot shows a Microsoft user profile page for 'Sarah Barrett'. At the top, there's a summary box with her name, email (sbarrett@microsoft.com), title ('Microsoft Employee'), and stats: 0 Following, 0 Followers, 13 Badges, 1 Trophy, 0 Answers accepted. To the right, it shows 'LEVEL 6' with a green diamond icon, 22000/32299 XP, and '1 reputation point earned.' Below this is a sidebar with links: Overview (which is selected), Activity, Bookmarks, Collections, Following, Achievements, and Settings. The main content area is titled 'Pick up where you left off' and lists three modules: 1. 'Assess your existing software development process' started 5 minutes ago, 21% completed, 15 min remaining. 2. 'Survey the services on the Azure Data platform' started 2 months ago, 8% completed, 53 min remaining. 3. 'Get data with Power BI Desktop' started 2 months ago, 13% completed, 1 hr 5 min remaining. At the bottom of this section is a link 'Browse all activities >'. Below this is a 'Certifications' section with a progress bar and the text 'Getting certified?'. At the very bottom, there's a URL bar showing 'https://docs.microsoft.com/en-us/learn/modules/assess-your-development-process/'.

A person who has created a profile. They may earn achievements, create content, complete self-assessments and recertification exams, and contribute questions/answers/comments.

## Live Example

*No current example*

## When to use Users

You need to represent a person on the site in some way. So that a person can track their activity and be identifiable on the site in some way.

## Elements of Users

- Affiliation
- Email Address
- Group
- Image
- Name
- URL

- XP
- Username
- Reputation

### **Taxonomies used on Users**

- Product
- Dev Language
- Role
- Level
- Subject

### **Relationships with other content types**

- Answers this user has provided
- User whose certification this is
- User who provided this answer
- Users participating in this challenge
- Users who subscribe to this show
- User who created this collection
- Questions this user has asked
- Users registered for this event
- User who initiated this assessment session
- Achievements this user has earned
- Users in this group
- Tags this user is following
- Assessment sessions this user has in progress
- Renewal assessments this user has completed
- Self-assessments this user has completed
- Episodes in this user's queue
- Collections this user has created
- Challenges this user has participated in
- Episodes in this user's history
- Shows this user is subscribed to
- Comments this user has provided
- User who provided this comment
- User who asked this question
- Sessions in this user's history
- Modules this user has in progress
- Group this user is in
- Reports this user has generated
- Certifications this user has
- Challenges this user is currently participating in

## **User Certification**

### **Status: Live**

The specific instance of a certification that a user has earned. There is a generic "Dynamics 365 Finance Functional Consultant Associate" certification, and then there is Jane Doe's Dynamics 365 Finance Functional Consultant Associate certification, that she got by taking Exams MB-300 and MB-310 in June 2020 and expires in June 2021 (a user certification).

## **Live Example**

*No current example*

## **When to use User Certifications**

User certifications are automatically-generated, not authored. Choose it when you need to represent an individual user's certification instance.

## **Elements of User Certifications**

- End Date
- Start Date

## **Taxonomies used on User Certifications**

- none

## **Relationships with other content types**

- User whose certification this is
- Renewal assessment that would extend this user certification
- Certification this user certification corresponds to
- Certifications this user has

# Unified Content Model FAQ

11/2/2020 • 6 minutes to read

## What is the unified content model?

Content models define the type of content within a system and the relationships between those types. The unified content model will design, implement, and maintain a shared content model. This shared content model will ensure all content on Docs.microsoft.com is structured.

## Does this include Learn? What about Q&A? Assessments?

The unified content model includes everything on Docs.microsoft.com. Our goal is to build an experience that brought together the various modalities that we're responsible for into a better, "One Microsoft" view of our content.

## Are we considering using X technical solution?

We're making sure new work aligns with the model and we're doing user research to make sure we're headed in the right direction. We haven't decided on specific technologies.

## What was the scale of the docsets the content model was developed on? Do we know that it works with the size of some of our docsets?

We developed the content model by thinking of all the content on Docs, Learn, Q&A, and in other neighborhoods as one large docset. Research indicates our customers need to experience it as one body of content. The authoring and content management workflow hasn't been designed yet, but we haven't forgotten about it.

## Which partners did we work with to understand their current content types and migration plan?

Initial vision: C&L Docs, Learn, Azure Architecture Center, the former Ops & Governance team.

First round of review: Business Applications Group, Microsoft 365 Dev content, Microsoft Graph, Microsoft 365 Commercial content, Protocols, .NET Mobile, and Azure Active Directory

Second round of review: Azure Active Directory, Cloud Adoption Framework, Dynamics, Microsoft Graph, Microsoft 365 Dev content, Microsoft 365 Commercial content, Protocols, Troubleshooting, and Xamarin.

## The world and user expectations change over time, is there any risk that by the time we implement the content model will be outdated?

We'll need to work iteratively and be flexible about the new content model. It will change frequently to support user and business needs. To support this rate of change, we should invest in platform fundamentals.

Implementing the content model will give us a better idea of what we have and let us do more with it. If we need to move in a different direction, it will be easier to do than it is now.

## Is there a mockup of what this will look like?

There aren't any final designs for the future experience. All implemented content types have screenshots in the

[documentation for the unified content model.](#)

## Isn't structured content hard to create? How will PMs and Devs write it?

It's difficult to create structured content in YAML, but anybody who's ever written an email has also created structured content. Structured content doesn't have to be difficult. We believe there are ways to make structured content easier to create than our current unstructured content is. To make that happen, we should invest in the authoring experience as an organization.

## What impact will the content model have on SEO?

Structuring content according to standard schemas is great for SEO and makes Google show much richer information in the SERP. We will be integrating standard schemas for all content types that have them.

## Will we have content types for code samples?

We do right now, for the [code samples on Docs](#). Making these code samples fully functional will require additional feature work. From a content model perspective, they are already live.

## Will we have content types for videos?

Possibly, when the feature work that requires them is prioritized. Videos don't need to be content types to be used in units, articles, or within other content types. We would add them as a content type if we wanted to develop experiences that are unique to videos.

## Will we have content types for images?

Probably not. There aren't as many likely use cases for treating images like individual content types. We would benefit from a digital asset management (DAM) tool for image management, but the content model can't provide that.

## Will we have a content type for navigation/TOCs?

We have some content types that are used for grouping other content, like learning paths and collections. We would like to eventually replace TOC-based navigation with navigation driven through these more flexible groupings. This work isn't planned yet, however.

## What does the content model have to say about article length?

The content model doesn't provide guidance about what makes content good, just what makes it the right type of thing. To use a silly analogy, you can build an ugly, unstable table that is still demonstrably a table. The content model cares whether something is a table; not whether it's a good table.

That said, moving to more structured content will make it easier for teams to define those content standards and assess performance and compliance.

## How will this handle content that exists in multiple versions through tabs and zone pivots?

Versioning is a known, but not solved, problem. Better structured content will give us more options to deal with problems like this, but we haven't tackled it yet.

## What testing have you done?

We did baseline testing with four realistic tasks on the live version of Docs with UserTesting.com. We then tested the same four tasks with clickable InVision prototypes. These prototypes were meant to be illustrative of one way the experience could work, not a definitive design for how it will work. Our goal was to test whether some of the more radical changes would be helpful to users. This round of research confirmed that they would.

## What work does my team have to do? How is content going to get migrated?

The basic cadence we're planning is:

1. Design a new content type. This work needs to be prioritized by the product team, it needs a spec, design, and dev time allocated.
2. Identify existing content that should be this new content type. Some content might be net new, like Questions and Answers in Q&A, while others, like FAQs, already exist on Docs.
3. Develop a migration strategy. Migration is likely to vary quite a bit depending on the new design and the content type. Some migrations may be able to be entirely automatic, while others will require manual work. This stage is where content teams will work with PMs to define that migration strategy.
4. Deploy the new content type. At this point, it will be available for new content being published.
5. Migrate existing content.

No content model-related content migrations are planned as of July 2020.

## Are we really getting rid of the TOC?

The current TOC is difficult to manage and doesn't do much of what we need it to. We'll introduce new, better navigational affordances over the course of FY21. In FY22, we'll evaluate whether the TOC is still needed.

## What are the plans to assess the adherence to the Content Model and sustaining it?

Adherence to the content model will be enforced at the platform level, much like it is for Learn or Q&A content right now. We don't assess whether Learn is adhering to the content model, because there are no other options than to create units, learning paths, and modules.

## How can I get involved?

We send out biweekly updates, which you can [sign up for](#). We also hold monthly Q&A sessions about the contents of those updates. Email [Sarah Barrett](#) for the invitation to those meetings.

# Breadcrumbs guidelines

11/2/2020 • 6 minutes to read

## The role of breadcrumbs

Breadcrumbs are a set of links at the top of a page, which convey its position in the overall site hierarchy. They allow a user to navigate through the hierarchy all the way back to the homepage, one level at a time, by starting at the last link in the breadcrumb trail.

Breadcrumbs are a recognized, conventional user interface (UI) pattern that's good for usability and SEO. They serve the following purposes:

- Users look for breadcrumbs to understand where they are.
- Users look at breadcrumbs to determine what they are looking at, the structure of the content set they're in, and what conceptual path they are currently on.
- Breadcrumbs provide pathways out and up from a page deep within a site.
- Breadcrumbs give users a way to "dock", or slowly move in closer to the content they need.

Breadcrumbs are commonly considered a best practice because they:

- Help draw in users who enter on deep pages that aren't well surfaced in navigation
- Are a standard pattern that, when applied consistently, never cause usability problems and can sometimes provide an escape valve for otherwise poorly designed experiences
- Take little space on the page and are generally low-impact to maintain
- Can be used by Google in search engine results pages (SERPS), making the result more human-readable. This can result in higher click-through-rates and lower page bounce rates.

## Wayfinding benefits of breadcrumbs

Additionally, improvements to the overall wayfinding on a site are important. When we say "wayfinding" we mean the user's ability to know where they are on the site, where they are trying to go, and how to get there from their present location.

To give users this knowledge, we want to ensure they have four kinds of paths available to them:

1. Top down – Paths that drill "down" from general to specific, like top hierarchical navigation.
2. Bottom up – Paths that zoom out from specific to general, like breadcrumbs.
3. Teleport – Paths that transport users directly between related articles, without interacting with a hierarchy, like "Customers who bought this item also bought".
4. Conversational – Paths that users activate by asking for help, like a search box.

Breadcrumbs are one of the primary ways we give users that ability to easily back out of a specific area. They also allow users to see through multiple layers of site hierarchy, back to the landmark that is the homepage. This is invaluable in communicating their present location and helping users intuit the rules of the site, which in turn, allows them to know their desired location and formulate a path for getting there.

## Best practices for breadcrumbs

### Semantic coherence

Breadcrumbs must be used **coherently**, in a way that adds meaning to the user's experience, rather than confuses it. Breadcrumbs that pop in and out of existence, lead to un-intuitive places, mislead users about the

actual structure of a site, or break established hierarchical patterns all lead to an incoherent, difficult to use experience. Speaking generally, coherence results from ensuring that content and structure is correct, clear, and consistent, in order of decreasing importance:

- **Correct:** Don't lie to users or represent things incorrectly, even if it would make the experience more consistent. Authors are often tempted to add additional layers of breadcrumbs to make a page look like it's located in the same place as other similar ones. If that's not actually the case, lying to the user doesn't improve the experience.
- **Clear:** Focus on conveying true information clearly, then on how to do that consistently, rather than indiscriminately applying confusing patterns. Breadcrumbs are a utilitarian affordance, and Docs is a utilitarian space.
- **Consistent:** When we can represent the path in a way that is correct and clear, it is then important to focus on doing so consistently across product families, so that users can predict how breadcrumbs will work everywhere.

### Placement and application

Breadcrumbs should be placed above the page title, in the same place on every page. They should always be accessible, but not intrusive. Most pages should have breadcrumbs. The exceptions to this are:

- L1 pages: Docs Homepage, Learn homepage, Q&A homepage, Code Samples homepage (browse)
- L2 pages: Hub pages, Learn browse pages Certification browse page, Certification overview, Learn TV page
- Search results pages
- Pages nested in interactive experiences, such as unit pages within modules, question pages in assessments, and so on.

### Mobile usage

Breadcrumbs need to be shortened for mobile display. The best practice is to show the level above the current document with an arrow pointing backwards, like "< Virtual Machines" in the same location.

## How to semantically structure breadcrumbs

This guide proposes an extensible breadcrumb structure appropriate for each neighborhood of Docs (Documentation, Learn, Q&A, and Code Samples, today). There are a few guidelines that apply to all breadcrumbs, everywhere on the site:

- Breadcrumbs should reflect the underlying model of the neighborhood in which the content exists. So, 'Documentation' may use one structure, while 'Learn' may use a different structure. All breadcrumbs should adhere to the established structure for the neighborhood in which they exist.
- Pages with multiple dynamic states (like tabs) don't need to have different breadcrumbs for each of these states.
- The breadcrumbs reflect the structural position of the page in the hierarchy, not the URL path or the user's path.

#### NOTE

Note: For Docs, each family (or repo, TBD) will be assigned an initial breadcrumb path as part of onboarding. The remainder of each document's breadcrumb path will be constructed based on its placement within the TOC. Individual authors should not be able to author custom breadcrumb paths.

### Breadcrumbs in Documentation

Structure: Content type > Parent product > Subproduct > Page title

Example in practice: Documentation > Visual Studio > VSTS > Git Repositories

Let's unpack this:

- The first unit or crumb is content type or "neighborhood" as represented in our header navigation, in this case, 'Documentation'.
- The second crumb should be the parent product for the content.
- The third crumb should be the subproduct or service, if applicable. Use up to two subproduct or subservice units.
- The final crumb of the breadcrumb should be the page title of the page you are on (not hyperlinked)
- A user should be able to navigate to the subsequent page from each intermediate page. In this case, VSTS should be linked on the Visual Studio page, and Visual Studio should be linked from the Docs homepage.

## Breadcrumbs in Learn

Structure: Content type > Modality name > Item name > Subitem name

Let's unpack this:

- First unit, content type = "Learn", because Learn is the neighborhood we are in.
- Second unit, modality name = Learning Paths & Modules, Certifications, or Learn TV
- Third unit, item name = the Learning path name if on a learning path page; the module name if on a module or unit page; the certification name if on a certification page; the video name if on a Learn TV video page
- Fourth unit, subitem name = should only be used for units, should be the unit name

### For learning paths

Structure for learning paths: Content type > Modality name > Learning path name

Example Learning path: Learn > Learning Paths & Modules > Azure fundamentals

### For modules and units

Structure for learning paths: Content type > Modality name > Module name > Unit name

Example module: Learn > Learning Paths & Modules > Predicts costs and optimize spending for Azure

Example unit: Learn > Learning Paths & Modules > Predicts costs and optimize spending for Azure > Introduction

### For certifications

Structure for certifications: Content type > Modality name > Item name

Example certificate: Learn > Certifications > Exam 77-81: Word 2010

## Breadcrumbs in Q&A

Structure for Q&A: Content type > Modality > Item name

Let's unpack this:

- Content type again is the neighborhood of Docs we are in. Here, it is Q&A.
- Modality options for Q&A are: questions, tags, or articles

For example:

- Question example: Q&A > Questions > Removal of incoming email attachments
- Tag example: Q&A > Tags > azure-cosmos-db
- Article example: Q&A > Articles > Verified answers

## Breadcrumbs in Code Samples

Structure for code samples: Content type > Code sample name

Example: Code Samples > Azure HDInsight FQDN lists

Note: Code samples are a hierarchically shallow experience compared to the rest of Docs, so their breadcrumbs are much shorter. Their content type is always 'Code Samples'.

# TOC structure

5/10/2021 • 5 minutes to read

Standard Content & Learning Tables of Contents (TOCs) share some common elements that create a sense of consistency across the documentation, which helps users navigate the site with ease. These elements are explained below.

For information about how to create and manage TOCs, see [TOC file formats](#).

## TOC goals

Consistency among our TOCs is important. Customers are learning our content model, and research suggests that the model is helpful when applied correctly. When customers recognize our content structures, it is easier for them to travel through our documentation to find the information they need.

Goals for a TOC:

- Present a useful amount of content, while still staying usable.
- Present content in a way that resonates with the customer's likely use cases for a product or technology.
- Allow users to move easily between topics, including rapid zooming in and out.
- Help users form helpful mental models by communicating a useful, structured perspective on how a product is organized.

## How many TOCs does my product need?

Most services have one TOC. For a larger product with multiple technologies, it might work better to have multiple TOCs, and then tie the TOCs together by linking to them from a product hub page. Deciding whether to keep all the technologies in the same TOC or break them up into multiple TOCs depends on these factors:

- Complexity of a product.
- Number of files per TOC. Each TOC should have enough breadth to allow a user to meaningfully use a product or accomplish an end-to-end task. In some larger products, there might be several subsections where users can accomplish significant tasks without interacting with the other sections.
- Amount of shared content. If there is considerable shared content between the technologies, one TOC is a better choice. For example, a Cosmos DB research study showed that users would rather have more information in one larger TOC than to bounce around to multiple smaller TOCs.

### Single TOC

A **standard single TOC** is one table of contents with one landing page. TOC categories are the top-level (L1) nodes and include: Overview, Quickstarts, Tutorials, Samples, Concepts, How-to guides, Reference, and Resources. There should be no other nodes at the top level of the TOC. Any exceptions to this structure must be approved by a C+AI team M2.

The following TOC shows the layout of a TOC that has a top-level node for each of the standard content types.

Note these requirements:

- Top link is *Service* documentation. This links to the landing page (index.yml), which is the default view.
- When there is a Quickstarts section, it is expanded by default. If there is no Quickstarts section, Tutorials is expanded by default.
- The variations of a quickstart (portal, CLI, PowerShell) are listed consecutively.

If you need to include [general troubleshooting](#) or [problem resolution](#) information, add a **Troubleshoot** node and possibly a **Problem resolution** node:

```
<Service name> documentation

> Overview
> Quickstarts
> Tutorials
> Samples
> Concepts
> How-to guides
    > Troubleshoot
        General troubleshooting article
        > Problem resolution
> Reference
> Resources
```

For information about content types, see [Choose the correct content type for your article](#).

## Multiple TOCs

For larger products, having multiple TOCs connected to a central hub page works well to organize diverse content. Hub pages are collections of related services, products, or languages. You use hub pages to help your customers better understand the bigger picture, see how different components fit together, and quickly guide them to their area of interest. The hub page itself does not have a single TOC, but links to landing pages where a single TOC is focused around a specific topic or product.

The screenshot shows the Microsoft Azure Active Directory documentation hub page. At the top, there's a navigation bar with links for Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More, and a Free account button. Below the navigation is a breadcrumb trail showing 'Azure / Active Directory'. On the right side of the header are links for 'Contact Sales: 1-800-867-1389', 'Search', 'Portal', and user profile options like 'pr-en-us-7987', 'Share', 'Theme', and 'Sign in'. The main title 'Azure Active Directory documentation' is displayed in a large blue header. A sub-header below it states: 'Azure Active Directory (Azure AD) is a multi-tenant, cloud-based identity and access management service.' Below the header, there are several cards: 'OVERVIEW What is Azure AD?', 'WHAT'S NEW What's new in Azure AD?', 'HOW TO GUIDE Assign roles to users', 'HOW TO GUIDE Create a group and add members', 'CONCEPT Azure AD deployment checklist', and 'HOW TO GUIDE Add a subscription to your tenant'. The main content area is organized into four columns: 'Application management' (with links to What is single sign-on (SSO), Automatic user provisioning, Application Proxy for on-premises apps, and a 'See more' link), 'Authentication' (with links to How it works: Azure MFA, Azure AD self-service password reset, Azure AD password protection, and a 'See more' link), 'Business-to-Business (B2B)' (with links to What is Azure AD B2B?, Add guest users in the portal, B2B and Office 365 sharing, and a 'See more' link), and 'Business-to-Customer (B2C)' (with links to What is Azure AD B2C?, Create an Azure AD B2C tenant, Custom policies in Azure AD B2C, and a 'See more' link). Below these are two rows of smaller boxes: 'Conditional Access' (What is Conditional Access?), 'Developers' (About Microsoft identity), 'Device management' (What is device management?), and 'Domain services' (What is Azure AD Domain).

In the example above, the hub page is built around the Azure Active Directory, where individual landing pages with TOCs, such as Authentication and Domain services, are linked to from the hub. Multiple TOCs work well in this case as there is substantial content that is specific to different aspects of the product itself. The breadth of content available for Azure Active Directory is too much for a single TOC to handle well within our current standards.

You can [learn more about how to create a hub page here](#), or [contact the DevRel Information Architecture team](#) to discuss your options if you're not sure whether your content fits a single TOC or hub page design.

## TOC variations

If you are considering experimental TOC organization or categories, please review the [guidelines on requesting approval for exceptions and experiments](#).

## TOC text and links

This section explains how to structure the content and links in the TOC that lives underneath the top-level nodes. This guidance applies whether you are creating a single TOC or multiple TOCs connected to a hub page.

### Keep the same context

- All articles in your TOC should display that same TOC when clicked into by a user, even when you need to link to an article in another folder. For a customer, suddenly landing in a new TOC when they click a link can be disorienting. If you want to reuse content from another area, use the [contextual TOC feature](#).
- All links should go to an article, not another TOC. Hub pages are designed for linking to multiple TOCs.
- Clicking on the right-most breadcrumb should always take you to the current TOC you are in.

### TOC size

- Other than the Overview and Samples sections, try to keep each individual list to between 3 and 12 items. Fewer than 3, and it may not need to be a category. More than 12, and it becomes difficult to scan.

- Try to go no more than three levels deep.
- Expand the quickstarts by default when a user lands on the landing page. Having no nodes expanded means every TOC looks the same and doesn't give users the context they need.

### **TOC labels and links**

- TOC labels (link text) should be short but similar to the H1 of the article they point to.
- Parent nodes should expand and not be a link.

### **External links**

Links to external sites from a TOC must be in the **Resources** node. Links in all other nodes should be to content on docs.microsoft.com (DMC).

### **Things that cause confusion**

- Don't have more than one link to the same file in the same TOC.
- Don't have a link in a TOC that takes you to a different TOC than the one you're in. Users don't know if that link will take them to a single article or a different context.
  - If you need to link to an article in a different TOC, consider using a [contextual TOC](#).
- Don't stray from using the standard categories since users expect to see them at the highest level of the TOC. Even where alternative categories are used (as with nested or stacked approaches), make deliberate modifications to the standard categories if you have to, but don't abandon them.
- Don't duplicate information structure. Hub pages and landing pages should provide different information than what users can glean by looking at the top level of the TOC.

# TOC file formats

6/3/2021 • 4 minutes to read

A table of contents (TOC) is used to define the structure of a docset. TOCs for `docs.microsoft.com` should be created in YAML, but you may already have a TOC in Markdown. Markdown TOCs are deprecated and should be [converted to YAML](#).

A TOC visualizer can be found at the [TOC Helper](#).

## YAML TOC format

The YAML-based TOC format provides more functionality than Markdown, such as auto-expansion of TOC nodes and automatic contextual TOC query string generation.

To build your TOC with YAML, create a file named `toc.yml` (always lowercase). Let's look at the structure for a simple YAML TOC:

```
items:
- name: Tutorial
  items:
    - name: Introduction
      href: tutorial.md
    - name: Step 1
      href: step-1.md
    - name: Step 2
      href: step-2.md
    - name: Step 3
      href: step-3.md
```

The YAML document is a list of TOC elements, each of which minimally has a `name` and `href`. TOC nodes can also act as parents to other nodes. Here, the child TOC nodes are represented by a list called `items` and the parent TOC node may not have an `href`. If an `href` is added to a parent TOC node, Docs automatically adds a new child node underneath the parent with the parent's name and `href` value. Docs also removes the `href` from the parent. This action is to avoid parent nodes acting as both expanders and content pages.

Here's an explanation of all the properties available on a YAML TOC node:

- `name` (required) - A string name that is displayed for the TOC node. The name can't include a colon (`:`).
- `displayName` (optional) - Alternate search terms for TOC filtering. A string value that doesn't get displayed (yes, it's a poor name) but is searched, as is `name`, during TOC filtering. For multiple values, use a comma-separated list like `displayName: batch, asynchronous, off-line, offline`.
- `href` (optional) - The path the TOC node navigates to. Optional because nodes can exist just to parent other nodes, in which case they may not have an `href`.
- `uid` (optional) - An identifier for any reference documentation on the Docs site, for example: `System.String.uid`. Optional because nodes can point to parent or other nodes or a specific page.
- `items` (optional) - If a node has children, the children are listed in the `items` array. The child nodes have the same available properties as listed above.
- `expanded` (optional) - This property specifies if the node should be expanded by default when the TOC is loaded. Only *one* root-level node can be expanded on load. The default value is `false`. Only add `false` with value `true` if you want the node to be expanded.
- `maintainContext` (optional) - **Don't use.** This option is no longer fully supported by the platform. Use the

guidance in [How to make a contextual TOC](#) to maintain the context of your TOC. Maintaining the context is important when you link to content in another TOC.

**TIP**

- Check `toc.yml` syntax with a YAML validator like <http://www.yamlint.com/>.
- Add an `items:` tag as the first line in the file to appease the linter.

Before:

```
fundamentals > ! toc.yml > ...
1   |- name: .NET documentation
2   |- href: index.yml
3   - name: Get started
4   | items:
5   |   - name: Hello World
6   |   | href: ../../core/get-started.md
7   |   - name: Get started tutorials
8   |   | href: ../../standard/get-started.md
```

After:

```
fundamentals > ! toc.yml > [ ] items
1   < items:|
2   <   -- name: .NET documentation
3   |   | href: index.yml
4   <   -- name: Get started
5   |   | items:
6   |   |   -- name: Hello World
7   |   |   | href: ../../core/get-started.md
8   |   |   -- name: Get started tutorials
9   |   |   | href: ../../standard/get-started.md
```

Here's a larger example YAML structure, which includes more configuration options:

```

items:
- name: Dev Sandbox
  href: index.md
  displayName: Home
  pdf_name: foo
- name: Compare ASP.NET Core and ASP.net
  uid: choose-between-aspnet-and-aspnetcore
- name: Conceptual Pages
  expanded: true
  items:
    - name: Overview
      href: ./conceptual/index.md
    - name: Code Samples
      href: ./conceptual/code.md
    - name: Tables
      href: ./conceptual/tables.md
- name: Reference Pages
  items:
    - name: IDictionary
      href: ./reference/System.Collection.IDictionary.yml
    - name: String
      href: ./reference/System.String.yml
- name: Content Pages
  href: ./content/index.md
- name: Hub Pages
  items:
    - name: Card Gallery
      href: ./hubPage/cardGallery.md
- name: Landing Pages
  items:
    - name: Azure Architecture
      href: ./landingPage/azureCat.md
    - name: UWP
      href: ./landingPage/uwp.md
- name: Engineering Excellence
  items:
    - name: Environment Setup
      href: ./eeds/environment-setup.md
    - name: Template Docs
      href: ./eeds/docs/index.md
- name: Break Title Tests
  items:
    - name: System.Automation.String.Foo.Bar.Zip.Test()
      href: ./eeds/environment-setup.md
    - name: VMSScaleSets-AzureRmDiagnosticsDscFixUp
      href: ./eeds/docs/index.md

```

When you display this page, it expands the `Conceptual Pages` node of the TOC.

## IMPORTANT

In your `docfx.json` file, make sure to add `"**/*.yml"` as a content file type, if it isn't already present. Otherwise the build system won't pick up the YAML file.

```

"build": {
  "content": [
    {
      "files": [
        "**/*.md",
        "**/*.yml"
      ],
      ...
    }
  ]
}

```

# Tips to avoid common TOC issues

Don't include the root folder in the href:

- Good: marketplace/overview.md
- Not good: /docs/marketplace/overview.md

Don't begin with a slash:

- Good: collaborate/overview.md
- Not good: /collaborate/overview.md

## Nested TOCs

To nest a TOC within another TOC, set the `href` property to point to the `toc.yml` file that you want to nest. You can also use this structure as a way to "reuse" a table of contents structure in one or more TOC files.

Consider the following two `toc.yml` files.

```
toc.yml :
```

```
items:
- name: Azure overview
  href: azure-overview.md
- name: Extensibility
  href: extensibility/toc.yml
- name: Reference
  href: azure-reference.md
```

```
extensibility/toc.yml :
```

```
items:
- name: Extensibility overview
  href: overview.md
- name: Create an extension
  href: create-extension.md
- name: Troubleshoot
  href: troubleshoot-extensions.md
```

This structure renders as follows:

```
Azure overview
  Extensibility
    Extensibility overview
    Create an extension
    Troubleshoot
  Reference
```

When the user selects any link from the nested TOC, they remain in the containing TOC. If the Extensibility node linked to `extensibility/overview.md` instead of `extensibility/toc.yml`, selecting **Extensibility** would take the user to a new TOC (the extensibility TOC).

#### NOTE

If a customer arrives through a search engine result directly to an article in a nested TOC, the nested TOC won't be displayed, but instead the entire TOC will be displayed.

## Convert existing Markdown TOC to YAML

To convert your `toc.md` file to `toc.yml`:

1. [Create a user story](#) with Content & Learning Content Production Services by selecting **Hub/Landing Page, UHF, and TOC Conversion**.
2. Follow the template link under **TOC Conversion (Markdown to YAML) Requests** to create a new user story in Azure Boards.
3. Provide the information requested. Also indicate the node you want to be expanded by default. For Content & Learning services, the Quickstarts node should be expanded by default.
4. Save your user story.

# Set up a contextual TOC and breadcrumb

5/10/2021 • 12 minutes to read

This article explains how to set up your TOC and breadcrumb files so that when you link to articles in another folder in your repo or another docs.microsoft.com repo, you maintain the context of your TOC.

Normally, if you link from your TOC to content in other repos or other folders in your repo, the context changes to the TOC of the article you linked to. This TOC sends the user away from your content and into a different content set, and can disorient your users.

However, you can set up your TOC to link to content in other folders in your repo or other repos on docs from your TOC and maintain the context of your TOC for the user. Maintaining context is a user experience best practice.

This article explains how to maintain context when:

- You want to link between TOCs in the same repo on docs.microsoft.com.
- You want to link between TOCs in different repos on docs.microsoft.com.

You will need to create or modify three files:

- **/breadcrumb/toc.yml** - To add mapping that displays a custom breadcrumb for an article that is not in your folder.
- **toc.yml** - To add a link to a file in another folder or repo.
- **/context/context.yml** - [optional] To provide a short string to append to your URLs. This file is also helpful to maintain the branding in browser for your repo when you link to article in a different repo that has a different branding.

All three files are hosted in the repository and subfolder of the product TOC context you are trying to maintain (we will show this in more detail in the following steps). You can use a text editor, such as VS Code, to work on these files.

## NOTE

It is best to create and publish breadcrumb and context files first in a separate pull request before making your TOC changes. This is especially important if you are using links to another repo - your breadcrumb changes will not preview if those files are not already published. These files are not seen by users unless they are digging through the repo on GitHub. They don't do anything until you reference them with TOC changes.

## Build a breadcrumb file

The breadcrumb file creates the breadcrumb path that shows as links above the TOC. It is important to maintain the context of the original product documentation not only in the TOC but also in the breadcrumb links. This way, when a customer selects a contextually linked article, the breadcrumb maintains the contextual location and hierarchy.

Azure / Key Vault

Feedback Edit

Filter by title

Key Vault Documentation

Overview

About Key Vault

Quickstarts

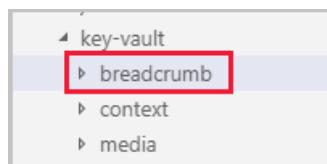
# What is Azure Key Vault?

07/16/2018 • 3 minutes to read • Contributors

Azure Key Vault helps solve the following problems

- Secrets Management - Azure Key Vault can be used to Securely store and tightly control access to tokens, passwords, certificates, API keys, and other secrets

Name the new breadcrumb file `toc.yml` and store it in a new subfolder called **breadcrumb** (in some repos, this folder is called **bread**). For example: `articles/key-vault/breadcrumb/` in the `azure-docs-pr` repo. This folder name is standardized across docs, so do not call it something else. If your product documentation has an existing breadcrumb folder with a TOC file, you can add an entry for a new mapping (explained in more detail below).



## Breadcrumb mapping between folders in the same repo

In this section, you will set up a breadcrumb file for linking to articles in the same repo. We're using `azure-docs-pr` and mapping between SQL Database and Key Vault.

For this example, we will link to an Azure SQL Database article from the Key Vault TOC. We want to maintain the Key Vault TOC and breadcrumb when the user clicks the link to the SQL DB content from the Key Vault TOC.

### Step 1. Find your index files

In your repo, find the index files for your service, and if needed, its grandparent. Documentation hub pages also have an `index.md` or `index.yml` file. You will use these URLs in your breadcrumb file to send customers to the right index page for that part of your breadcrumb.

- For Azure, the hierarchy `articles/index.md` becomes the URL <https://docs.microsoft.com/azure/>.
- For Key Vault, a service under Azure, the hierarchy `/articles/key-vault/index.yml` becomes the URL <https://docs.microsoft.com/azure/key-vault/>.

### Step 2. Gather a list of documents that you want to link to from your TOC

For linking to articles in the same repo, you need the location of the `.md` file and the URL (strip out the lang-locale). For example:

REPO	FILE LOCATION	URL
<code>azure-docs-pr</code>	<code>articles/sql-database/sql-database-always-encrypted-azure-key-vault.md</code>	<a href="https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault">https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault</a>

### NOTE

Context of your TOC will not be maintained if the article you want to link to from your TOC has metadata

`layout: ContentPage`

### Step 3. Build your breadcrumb file

Look at the Azure SQL Database article [Always Encrypted: Protect sensitive data in SQL Database and store your encryption keys in Azure Key Vault](#).

Note the structure of the URL after `docs.microsoft.com`: `/azure/sql-database/sql-database-always-encrypted-`

azure-key-vault. Normally, this structure creates a breadcrumb in Azure SQL Database of **Azure / SQL Database** where:

- **Azure** goes to <https://docs.microsoft.com/azure/index>
- **SQL Database** goes to <https://docs.microsoft.com/azure/sql-database/index>

If you want to add the Azure SQL Database article to the Key Vault TOC, you can create a mapping in the breadcrumb file that maintains the Key Vault context. This replaces the **/ SQL Database** breadcrumb link with a **/ Key Vault** breadcrumb link that maps to <https://docs.microsoft.com/azure/key-vault/>.

1. In your `/breadcrumb/toc.yml` file, create an element that sets Key Vault as the child breadcrumb to Azure:

```
- name: Azure
  tocHref: /azure/
  topicHref: /azure/index
  items:
    - name: Key Vault
      tocHref: /azure/sql-database/
      topicHref: /azure/keyvault/index
```

Lines 1-3 create a top-level breadcrumb element named Azure (name: Azure) and display it for any article with a URL that contains `/azure/` (tocHref: `/azure/`). When the user clicks the breadcrumb, they go to the `/azure/index` page (topicHref: `/azure/index`).

Lines 4-7 create a child breadcrumb element named Key Vault and display it for any article with a URL that contains `/azure/sql-database/`. When the user clicks the breadcrumb, they go to the `azure/key-vault/index` page.

2. Push your changes, submit a PR, review, and sign off. After the PR merges, the `.yml` file will be published as a `.json` file that you will point to later when you [build your TOC file](#). For this example:

FILE	FILE LOCATION	PUBLIC URL
Breadcrumb	articles/key-vault/breadcrumb/toc.yml	<a href="https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json">https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json</a>

At this point, you are ready to begin building or updating your TOC file.

### Breadcrumb mapping between different repos

In this scenario, you link from one repo to another, but the breadcrumb path and behavior persists as if you were in the original repo. We will use an example from Key Vault and Configuration Manager.

Follow steps 1 and 2 of [Breadcrumb mapping between folders in the same repo](#)

#### Step 3. Build your breadcrumb file

For this example, we will link to an article in the SCCM repo from the Key Vault TOC (in `azure-docs-pr`). We want to maintain the Key Vault TOC and breadcrumb when the user clicks the link to the Configuration Manager content from the Key Vault TOC.

Look at the article [Create and run PowerShell scripts from the Configuration Manager console](#).

Note the structure of the URL after `docs.microsoft.com:/sccm/apps/deploy-use/create-deploy-scripts`. Normally, when you open this article, the breadcrumbs in SCCM are **Docs / Enterprise Mobility + Security / Microsoft Endpoint Manager / Configuration Manager / Application management** where:

- **Docs** goes to the Docs hub page <https://docs.microsoft.com/>.
- **Enterprise Mobility + Security** goes to the EMS hub page <https://docs.microsoft.com/enterprise->

[mobility-security/](#).

- **Microsoft Endpoint Manager** goes to the Endpoint Manager hub page <https://docs.microsoft.com/en-us/mem/>.
- **Configuration Manager** goes to the Configuration Manager hub page <https://docs.microsoft.com/sccm/>.
- **Application management** goes directly to the article under [apps/understand](#) <https://docs.microsoft.com/sccm/apps/understand/introduction-to-application-management>.

SCCM is a repo that has a different content model and different branding than Azure Key Vault. Additionally, SCCM breadcrumb behavior clues us in on two things that might be useful later, depending on your scenario:

- Sometimes there isn't a landing or a hub page for sections.
- Breadcrumb links can be sent to a particular article, if needed.

To maintain the Key Vault context when linking to an SCCM article from the Key Vault TOC, we'll configure three breadcrumb layers in the breadcrumb file:

- Any article with `/sccm/` in the URL will display the top level breadcrumb node **Azure** that goes to the Azure hub page.
- Any article with `/sccm/apps/` in the URL will display the node **/ Key Vault** and map it to <https://docs.microsoft.com/azure/key-vault/index>.
- Any URL with `/sccm/apps/deploy-use` in the URL will display the node **/ Run SCCM scripts** and will also be mapped to the same Key Vault page.

The following YAML creates this breadcrumb structure:

```
- name: Azure
  tocHref: /sccm/
  topicHref: /azure/index
  items:
    - name: Key Vault          # Original doc set name
      tocHref: /sccm/apps/     # Destination doc set route
      topicHref: /azure/key-vault/index   # Original doc set route
      items:
        - name: Run SCCM scripts # Destination doc set name
          tocHref: /sccm/apps/deploy-use/ # Destination doc set route
          topicHref: /azure/key-vault/index # Original doc set route
```

Check the alignment and spacing of items in the breadcrumb file:

- The dash before name needs to line up directly under the "I" in items.
- The "t" in tocHref needs to be directly under the "n" in name.
- You need a space between the hyphen and name, and another space after the colon and before the field value.
- Make sure your fields are correctly spelled and that you end your topicHref value with /index.

Push your changes, submit a PR, review, and sign off. After the PR merges, you're ready to begin building or updating your TOC file.

#### IMPORTANT

YAML files are white space sensitive. An extra or missing a space will cause the build to fail.

**TIP**

Check the syntax with a YAML validator like <http://www.yamllint.com/>

## Build a TOC file

### Link to files in other folders in same repo or in a different repo

1. Gather the list of articles you want listed in your TOC. For articles in the same repo, note their relative links. For articles in a different repo, note the full public URL (without locale). For example:

REPO	FILE LOCATION (SAME REPO ONLY)	PUBLIC URL (WITHOUT LOCALE)
SCCMdocs-pr		<a href="https://docs.microsoft.com/sccm/aps/deploy-use/create-deploy-scripts">https://docs.microsoft.com/sccm/aps/deploy-use/create-deploy-scripts</a>
Azure-docs-pr	articles/sql-database/sql-database-always-encrypted-azure-key-vault.md	<a href="https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault">https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault</a>

2. Gather the full public URLs (without locale) of your breadcrumb TOC file and your documentation TOC file (in your repo/folder). You will need them for testing later. For example:

FILE	FILE LOCATION	PUBLIC URL
Breadcrumb	articles/key-vault/breadcrumb/toc.yml	<a href="https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json">https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json</a>
TOC	articles/key-vault/toc.yml	<a href="https://docs.microsoft.com/azure/key-vault/toc.json">https://docs.microsoft.com/azure/key-vault/toc.json</a>

3. Using the public URL for your TOC file, grab everything after the end of docs.microsoft.com. In this example: `/azure/key-vault/toc.json`. This is the forced TOC path.

The forced breadcrumb path is `/azure/key-vault/breadcrumb/toc.json`.

To get an idea of what it will look like, you can test your forced context on any docs.microsoft.com article. Just append the forced paths at the end of any article's URL in the browser as follows:

`?toc=<forced TOC path>&bc=<forced breadcrumb path>` . You should see the breadcrumb and TOC change to your product documentation.

In our example: `?toc=/azure/key-vault/toc.json&bc=/azure/key-vault/breadcrumb/toc.json`

4. Now you are ready to update your TOC file. Find the link you want to maintain the context for and append it with the forcing syntax.
  - a. For links internal to your repo, reference them like you normally would (use the relative path). Append the end with `?toc=<forced TOC>&bc=<forced breadcrumb>` . Here's a TOC YAML example:

```
- name: Secure SQL data
  href: ../../sql-database/sql-database-always-encrypted-azure-key-vault.md?toc=/azure/key-vault/toc.json&bc=/azure/key-vault/breadcrumb/toc.json
```

- b. For links outside your repo, use **Site Relative URL**, with domain name and language codes stripped out. Append `?toc=<forced TOC>&bc=<forced breadcrumb>` to the end. Here's a TOC YAML example:

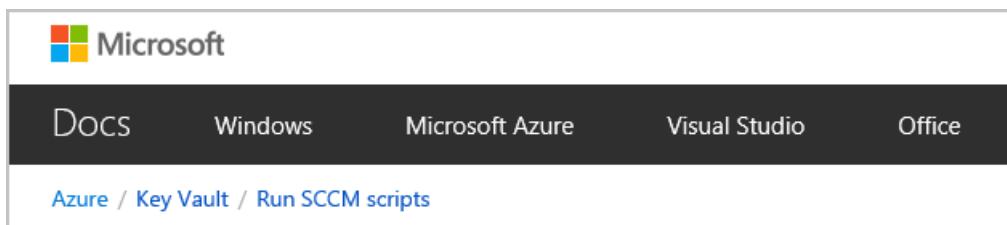
```
- name: SCCM Run Scripts
  href: /sccm/apps/deploy-use/create-deploy-scripts?toc=/azure/key-vault/toc.json&bc=/azure/key-vault/breadcrumb/toc.json
```

#### IMPORTANT

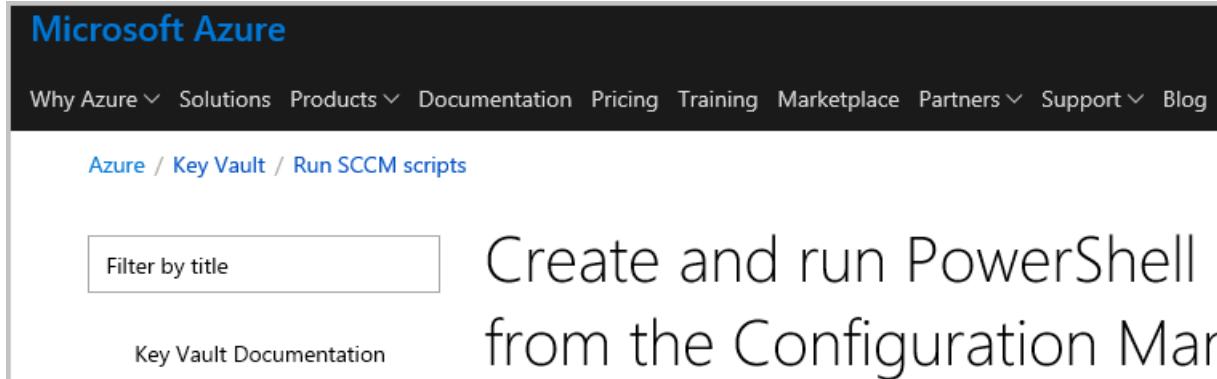
- The `href` value is not case-sensitive.
- The `href` value should not include the language code, `/en-us/`.
- DO NOT USE `maintainContext: true`.

## Build a context file

By using the modified TOC and breadcrumb file, you force what TOC and breadcrumbs will appear when the user follows the link. But when you link to articles in different repos, you may also need to deal with a different chrome. For example, System Center Configuration Manager carries the following branding:



While the Azure header (or "chrome") looks like



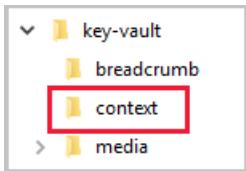
You can force the chrome appearance as well by using a context file. It is a user experience best practice to maintain a consistent chrome along with the TOC and breadcrumb.

The context file contains references to your breadcrumb file and your documentation TOC, and can identify the chrome to use.

#### NOTE

If you aren't linking to files in other repos, or if the chrome doesn't change for those repos, you don't need a context file. However, you may prefer to use it as it is a shorter string to append to links in TOC.

1. Create a new context YAML file (for this example, we will name the file `kv-context.yml`) and store it in a new folder called **context**. Like the **breadcrumb** folder, this folder name is standardized across docs and should be a subfolder of your product's content folder.



2. Identify the brand you want to apply. Typically, brand is defined in the docfx.json file for your repo. In our example, the brand is Azure and we want to maintain the Azure chrome.

```
! toc.yml • ! context.yml docfx.json x
69 "feedback_product_url": "https://feedback.azure.com/forums/341
70 "breadcrumb_path": "/azure/bread/toc.json",
71 "brand": "azure",
72 "searchscope": "Azure".
```

3. In your context.yml file, add a comment that this is a context object. Then define the brand that you found in the docfx.json file.

```
### YamlMime: ContextObject
brand: azure
```

4. Add two lines as shown below with relative links to the breadcrumb TOC file you created and your documentation TOC.

```
### YamlMime: ContextObject
brand: azure
breadcrumb_path: ../breadcrumb/toc.yml
toc_rel: ../toc.yml
```

5. Publish the context file. Once it publishes, verify it exists as a json file:

<https://docs.microsoft.com/azure/key-vault/context/kv-context.json>

6. After you verify the context JSON file, update your documentation TOC to use the shorter contextual link. You no longer need to append the forced breadcrumb and toc paths to the link. Instead, add the shortened context string, for example: ?context=/azure/key-vault/context/kv-context. This seems to work both with or without the forward slash after the equal sign. Here's a TOC entry example:

```
- name: SCCM Run Scripts
  href: /sccm/apps/deploy-use/create-deploy-scripts?context=/azure/key-vault/context/kv-context
```

7. Submit a new PR for this documentation TOC change to test the new contextual link. When testing the link, be aware that once clicked and the new link has resolved in the browser you will no longer be in your branch. Any other new contextual links will not work unless you go back to your branch. For each new link that you need to test, you will need to go back to your branch before testing.

#### NOTE

There appear to be some limitations with a context file when forced into a specific view such as ?view=azurermps-6.2.0 (for example PowerShell and CLI versions). This is a moniker and they already have conceptual mapping to certain TOCs.

```
① monikerMapping.json *  
1  | Qinen Zhu, 8 months ago • Restructure repo (#451)  
2  | "azurermps-6.2.0": {  
3  |     "serviceMap": "mapping/rm-groupMapping-6.2.0.json",  
4  |     "packageRoot": "azurermps-6.2.0",  
5  |     "conceptualToc": "docs-conceptual/azurermps-6.2.0/toc.yml",  
6  |     "conceptualTocUrl": "/powershell/azure/toc.json",  
7  |     "referenceTocUrl": "/powershell/module/toc.json"  
8  | },  
9  | "azurermps-5.7.0": {
```

In these cases, use your forced breadcrumb and TOC paths rather than a context file. For example:

```
/cli/azure/keyvault/certificate?toc=/azure/key-vault/toc.json&bc=/azure/key-vault/breadcrumb/toc.json#az-keyvault-certificate-create
```

## Report a bug

If you notice inconsistent behavior when you apply the contextual TOC, it may be a bug. If you want to report inconsistent behavior or have other supportability issues, raise your question in the [Docs support channel](#).

# TOC Visualizer

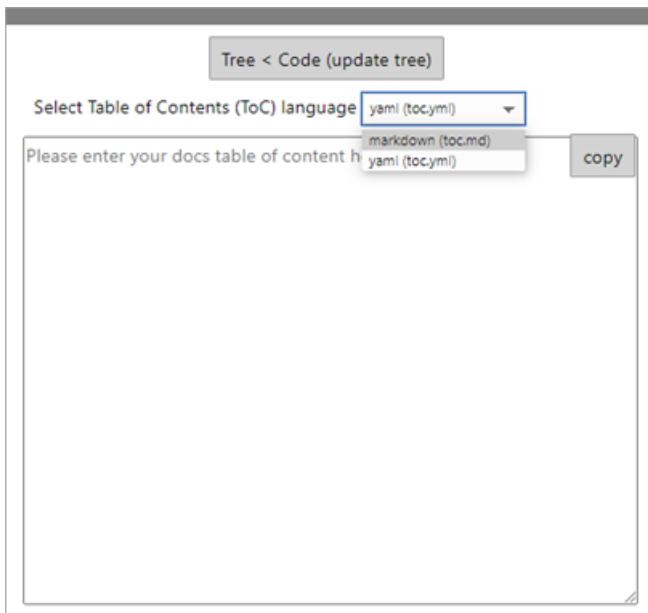
6/16/2021 • 2 minutes to read

To better help authors work with their Tables of Contents (TOCs), we've created a web-based tool to help manage and view the TOC.

You can find the Visual Studio Code extension in the VisualStudio Marketplace: [Docs Table of Contents Visualizer](#). And here's where you can find the web-based tool: [TOC Visualizer](#).

## Convert TOCs from Markdown to YAML

You can load a Markdown-based TOC into the tool. Once you've loaded your code into the TOC Visualizer, you can *only* convert the finished product to YAML.



### NOTE

You can't generate a new Markdown TOC.

## Manage entries

This tool lets you create, delete, and edit the TOC entries.

### Create a new TOC entry

To add a new TOC entry, select **Manage Tree [+]** > **Create Node [+]**. The **ADD NEW NODE** section appears:

**ADD NEW NODE**

Node Title   
 Additional Terms   
 Node href   
 Node uid   
 ToC Href   
 Topic Href   
 Maintain ToC Context

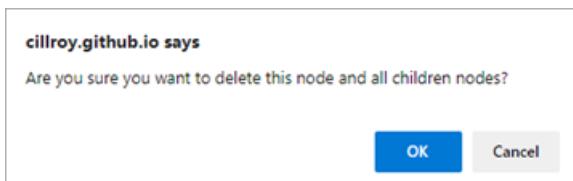
### Delete an existing TOC entry

To remove a TOC entry, select **Manage Tree [+]**, select the node you want to remove, and select **Delete Node**:

**Delete Node** **Create Node [+]** **Expand all nodes +** **Collapse all nodes -**

- Node 1 ([node1.md](#))
- \*  Folder 2 ([node2.md](#))
  - Node 2.1 ([node2.1.md](#))
  - Node 2.2 ([node2.2.md](#))
- \*  Folder 3 ([node3.md](#))
  - Node 3.1 ([node3.1.md](#))

A dialog box appears asking you if you're sure you want to delete the node and its children:



### Edit an existing TOC entry

To edit a TOC entry, select **Manage Tree [+]** and select the node you want to edit. The node's details populate the **Edit Selected Node** section:

**Edit Node [-]**

**Edit Selected Node**

Node Title    
 Additional Terms   
 Node href    
 Node uid   
 ToC Href   
 Topic Href   
 Maintain Context for Shared ToC

## Drag entries

You can drag a node to any position within the TOC structure:

The screenshot shows the TOC Visualizer interface. At the top, there are buttons for 'Manage Tree [-]', 'Tree > Code (generate YAML)', and 'Reset to Sample Data'. Below these are search and filter fields. On the right, there's a button 'Tree < Code (update tree)' and a dropdown for 'Select Table of Contents (ToC) language' set to 'yaml (tocyaml)'. The main area contains a tree view with the following structure:

- Node 1 ([node1.md](#))
- Folder 2 ([node2.md](#))
  - Node 2.1 ([node2.1.md](#))
  - Node 2.2 ([node2.2.md](#))
- Folder 3 ([node3.md](#))
  - Node 3.1 ([node3.1.md](#))

A node under 'Folder 2' has a blue background, indicating it is selected or being dragged.

## Sort TOC

You can sort the TOC at the branch level or sort the entire TOC:

The screenshot shows the TOC Visualizer interface. The tree structure is identical to the previous one, but the entire tree view is highlighted with a blue background, indicating it is selected for sorting.

## Support for advanced attributes

The TOC Visualizer provides support for advanced attributes.

### Experimentation ID

Select the Use Experimentation option and enter an Experimentation Id value.

The screenshot shows the TOC Visualizer interface. The 'Use Experimentation' checkbox and its input field are highlighted with a red border. The input field contains the placeholder 'Experimentation Id'.

When you generate your YAML TOC code, you'll see a `metadata` block at the beginning:

```
metadata:  
  experimental: true  
  experiment_id: "<your-experimentation-id>"  
  ...
```

## Additional terms for TOC filtering on Docs

Enter an Additional Terms value.

Edit Node [-]

### Edit Selected Node

Node Title  Update Node

Additional Terms

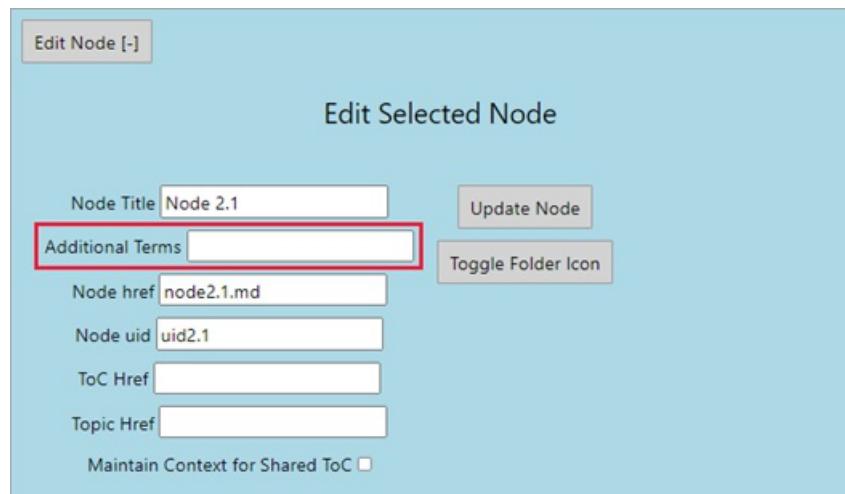
Node href  Toggle Folder Icon

Node uid

ToC Href

Topic Href

Maintain Context for Shared ToC



When you generate your YAML TOC code, you'll see a `displayName` value in the node block:

```
...  
- name: Node 1  
  displayName: <your-additional-term-value>  
  href: node1.md  
  ...
```

## Node UID

Enter a Node uid value.

Edit Node [-]

### Edit Selected Node

Node Title  Update Node

Additional Terms

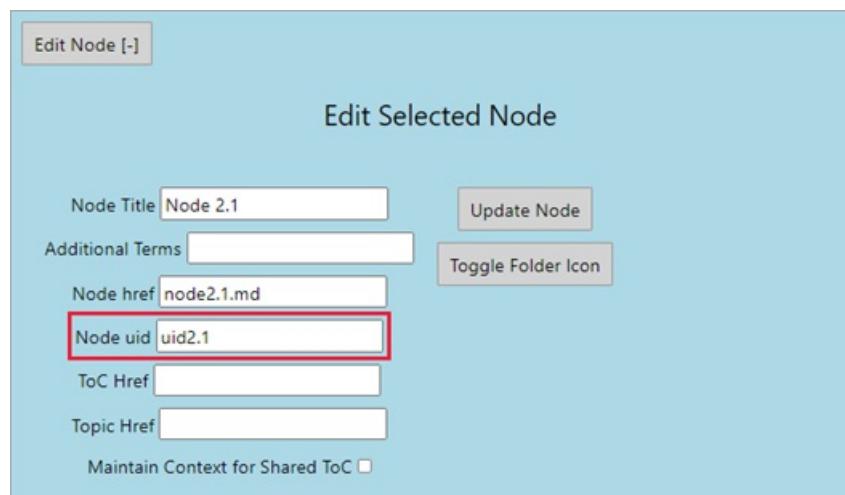
Node href  Toggle Folder Icon

Node uid

ToC Href

Topic Href

Maintain Context for Shared ToC



When you generate your YAML TOC code, you'll see a `uid` value in the node block:

```
...  
items:  
  - name: Node 2.1  
    href: node2.1.md  
    uid: <your-uid-value>  
  ...
```

## Maintain context for shared TOC

Select the **Maintain Context for Shared ToC** option.

Edit Node [-]

### Edit Selected Node

Node Title  Update Node

Additional Terms

Node href  Toggle Folder Icon

Node uid

ToCHref

TopicHref

Maintain Context for Shared ToC

When you generate your YAML TOC code, you'll see a `maintainContext` value of `true` in the node block:

```
...
items:
  - name: Node 2.1
    href: node2.1.md
    maintainContext: true
    uid: uid2.1
...
...
```

# Create or update a hub page

6/16/2021 • 24 minutes to read

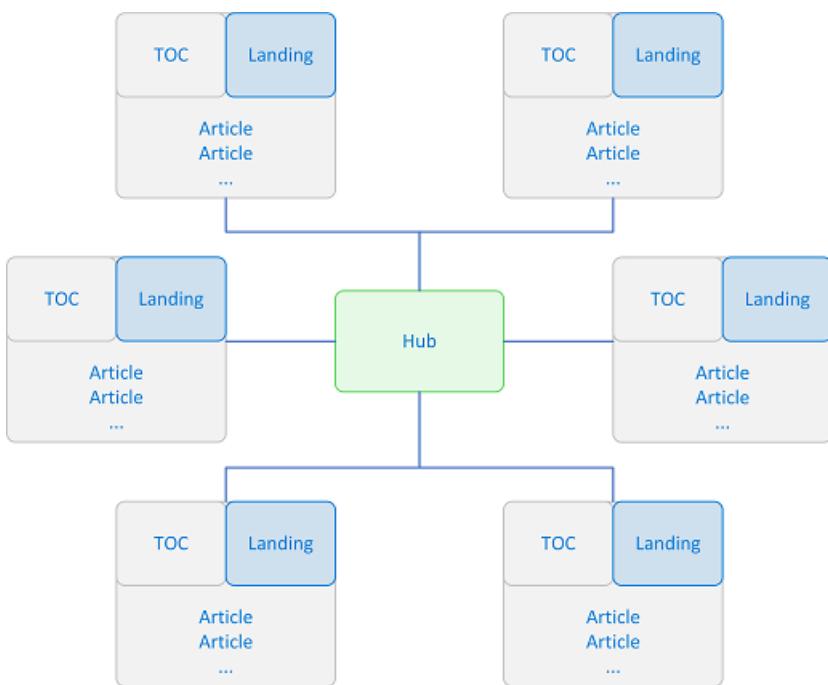
This topic is for Azure services and other products that follow the [Content & Learning content model](#). It describes the high-level process to create or update a hub page.

If you instead need to update the [Azure docs hub page](#), see [Update Azure docs hub page](#).

<https://aka.ms/docshub>

## Overview

Hub pages are collections of related services, products, or languages. You use hub pages to help your customers better understand the bigger picture, see how different components fit together, and quickly guide them to their area of interest. Hub pages are at an organizational level above [landing pages](#). The following diagram shows how you might use a hub page in your content structure.



Hub pages are different than landing pages in the following ways:

- Do not have a TOC
- Typically broader subjects or product areas
- Can link to landing pages or different content areas
- Design is more flexible and supports multiple sections

## Design

Starting in August 2019, a card-based hub page design was implemented that replaces the previous design. All teams in Content & Learning should use this new design going forward. This new design has the following benefits:

- Improve browsing experience for users to find content faster by creating a more dense presentation of info and revealing more content
- More focused on top customer tasks, products, or services

- Create consistent user experiences for hub and landing pages
- Support products of varying complexity and depth
- Simpler design

Here's an example of the **new** hub page design:

The screenshot shows the Microsoft Azure Active Directory documentation hub page. At the top, there's a navigation bar with links like Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More, and a Free account button. Below the navigation is a breadcrumb trail showing 'Azure / Active Directory'. On the right side of the header are links for 'pr-en-us-7987', 'Share', 'Theme', and 'Sign in'.

## Azure Active Directory documentation

Azure Active Directory (Azure AD) is a multi-tenant, cloud-based identity and access management service.

The main content area features several cards:

- OVERVIEW**: What is Azure AD?
- WHAT'S NEW**: What's new in Azure AD?
- HOW TO GUIDE**: Assign roles to users
- HOW TO GUIDE**: Create a group and add members
- CONCEPT**: Azure AD deployment checklist
- HOW TO GUIDE**: Add a subscription to your tenant

Below these cards are four larger sections:

- Application management**: What is single sign-on (SSO)?, Automatic user provisioning, Application Proxy for on-premises apps. [See more >](#)
- Authentication**: How it works: Azure MFA, Azure AD self-service password reset, Azure AD password protection. [See more >](#)
- Business-to-Business (B2B)**: What is Azure AD B2B?, Add guest users in the portal, B2B and Office 365 sharing. [See more >](#)
- Business-to-Customer (B2C)**: What is Azure AD B2C?, Create an Azure AD B2C tenant, Custom policies in Azure AD B2C. [See more >](#)

At the bottom, there are four additional sections:

- Conditional Access**: What is Conditional Access?
- Developers**: About Microsoft identity
- Device management**: What is device management?
- Domain services**: What is Azure AD Domain

Here's an example of the **previous** hub page design:

## Example hub pages

HUB PAGE	SOURCE
<a href="#">Azure Active Directory</a>	<a href="https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/active-directory/index.yml">https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/active-directory/index.yml</a>
<a href="#">Endpoint Configuration Manager</a>	<a href="https://github.com/MicrosoftDocs/memdocs/blob/master/memdocs/index.yml">https://github.com/MicrosoftDocs/memdocs/blob/master/memdocs/index.yml</a>
<a href="#">SQL</a>	<a href="https://github.com/MicrosoftDocs/sql-docs/blob/live/docs/index.yml">https://github.com/MicrosoftDocs/sql-docs/blob/live/docs/index.yml</a>
<a href="#">Visual Studio</a>	<a href="https://github.com/MicrosoftDocs/visualstudio-docs/blob/master/docs/index.yml">https://github.com/MicrosoftDocs/visualstudio-docs/blob/master/docs/index.yml</a>

## Hub page template

To create your hub page, you use YAML. The source file is an index.yml file typically in the root folder of the hub content. The following template defines the structure. Add this structure to the index.yml file. For a description of each section, see the sections later in this article.

```
### YamlMime:Hub

title: service documentation # < 60 chars
summary: summary # < 160 chars
# brand: aspnet | azure | dotnet | dynamics | m365 | ms-graph | office | power-apps | power-automate |
power-bi | power-platform | power-virtual-agents | sql | sql-server | vs | visual-studio | windows | xamarin
brand: brand

metadata:
```

```

title: title # Required; page title displayed in search results. include the drama. < 60 chars.
description: description # Required; article description that is displayed in search results. < 160 chars.
services: service
ms.service: service #Required; service per approved list. service slug assigned to your service by ACOM.
ms.subservice: subservice # Optional; Remove if no subservice is used.
ms.topic: hub-page # Required
ms.collection: collection # Optional; Remove if no collection is used.
author: githubauthor #Required; your GitHub user alias, with correct capitalization.
ms.author: msauthor #Required; microsoft alias of author; optional team alias.
ms.date: 01/27/2020 #Required; mm/dd/yyyy format.

# highlightedContent section (optional)
# Maximum of 8 items
highlightedContent:
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
# quickstart | reference | sample | tutorial | video | whats-new
  items:
    # Card
    - title: cardtitle1
      itemType: itemType
      url: file1.md OR https://docs.microsoft.com/file1
    # Card
    - title: cardtitle2
      itemType: itemType
      url: file2.md OR https://docs.microsoft.com/file2
    # Card
    - title: cardtitle3
      itemType: itemType
      url: file3.md OR https://docs.microsoft.com/file3

# productDirectory section (optional)
productDirectory:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      links:
        - url: file1a.md OR https://docs.microsoft.com/file1a
          text: linktext1a
        - url: file1b.md OR https://docs.microsoft.com/file1b
          text: linktext1b
    # Card
    - title: cardtitle2
      imageSrc: ./media/index/image2.svg OR https://docs.microsoft.com/media/logos/image2.svg
      links:
        - url: file2a.md OR https://docs.microsoft.com/file2a
          text: linktext2a
        - url: file2b.md OR https://docs.microsoft.com/file2b
          text: linktext2b
    # Card
    - title: cardtitle3
      imageSrc: ./media/index/image3.svg OR https://docs.microsoft.com/media/logos/image3.svg
      links:
        - url: file3a.md OR https://docs.microsoft.com/file3a
          text: linktext3a
        - url: file3b.md OR https://docs.microsoft.com/file3b
          text: linktext3b

# conceptualContent section (optional)
conceptualContent:
# Supports up to 3 sections
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
# quickstart | reference | sample | tutorial | video | whats-new
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:

```

```
# Card
- title: cardtitle1
links:
  - url: file1.md OR https://docs.microsoft.com/file1
    itemType: itemType
    text: linktext1
  - url: file2.md OR https://docs.microsoft.com/file2
    itemType: itemType
    text: linktext2
  - url: file3.md OR https://docs.microsoft.com/file3
    itemType: itemType
    text: linktext3
# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more
# Card
- title: cardtitle2
links:
  - url: file1.md OR https://docs.microsoft.com/file1
    itemType: itemType
    text: linktext1
  - url: file2.md OR https://docs.microsoft.com/file2
    itemType: itemType
    text: linktext2
  - url: file3.md OR https://docs.microsoft.com/file3
    itemType: itemType
    text: linktext3
# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more
# Card
- title: cardtitle3
links:
  - url: file1.md OR https://docs.microsoft.com/file1
    itemType: itemType
    text: linktext1
  - url: file2.md OR https://docs.microsoft.com/file2
    itemType: itemType
    text: linktext2
  - url: file3.md OR https://docs.microsoft.com/file3
    itemType: itemType
    text: linktext3
# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more

# conceptualContent section (optional)
conceptualContent:
# Supports up to 3 sections
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
quickstart | reference | sample | tutorial | video | whats-new
  title: sectontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      links:
        - url: file1.md OR https://docs.microsoft.com/file1
          itemType: itemType
          text: linktext1
        - url: file2.md OR https://docs.microsoft.com/file2
          itemType: itemType
          text: linktext2
        - url: file3.md OR https://docs.microsoft.com/file3
          itemType: itemType
          text: linktext3
```

```

# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more

# Card
- title: cardtitle2
links:
  - url: file1.md OR https://docs.microsoft.com/file1
    itemType: itemType
    text: linktext1
  - url: file2.md OR https://docs.microsoft.com/file2
    itemType: itemType
    text: linktext2
  - url: file3.md OR https://docs.microsoft.com/file3
    itemType: itemType
    text: linktext3

# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more

# Card
- title: cardtitle3
links:
  - url: file1.md OR https://docs.microsoft.com/file1
    itemType: itemType
    text: linktext1
  - url: file2.md OR https://docs.microsoft.com/file2
    itemType: itemType
    text: linktext2
  - url: file3.md OR https://docs.microsoft.com/file3
    itemType: itemType
    text: linktext3

# footerLink (optional)
footerLink:
  url: filefooter.md OR https://docs.microsoft.com/filefooter
  text: See more

# tools section (optional)
tools:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      url: file1.md
    # Card
    - title: cardtitle2
      imageSrc: ./media/index/image2.svg OR https://docs.microsoft.com/media/logos/image2.svg
      url: file2.md
    # Card
    - title: cardtitle3
      imageSrc: ./media/index/image3.svg OR https://docs.microsoft.com/media/logos/image3.svg
      url: file3.md

# additionalContent section (optional)
# Card with summary style
additionalContent:
  # Supports up to 4 sections
  sections:
    - title: sectiontitle # < 60 chars (optional)
      summary: sectionsummary # < 160 chars (optional)
      items:
        # Card
        - title: cardtitle1
          summary: cardsummary1
          url: file1.md OR https://docs.microsoft.com/file1
        # Card

```

```

- title: cardtitle2
  summary: cardsummary2
  url: file1.md OR https://docs.microsoft.com/file2
# Card
- title: cardtitle3
  summary: cardsummary3
  url: file1.md OR https://docs.microsoft.com/file3
# footer (optional)
footer: "footertext [linktext](https://docs.microsoft.com/footerfile)"

# additionalContent section (optional)
# Card with links style
additionalContent:
  # Supports up to 4 sections
  sections:
    - title: sectiontitle # < 60 chars (optional)
      summary: sectionsummary # < 160 chars (optional)
      items:
        # Card
        - title: cardtitle1
          links:
            - text: link1a
              url: file1a.md OR https://docs.microsoft.com/file1a
            - text: link1b
              url: file1b.md OR https://docs.microsoft.com/file1b
        # Card
        - title: cardtitle2
          links:
            - text: link2a
              url: file2a.md OR https://docs.microsoft.com/file2a
            - text: link2b
              url: file2b.md OR https://docs.microsoft.com/file2b
        # Card
        - title: cardtitle3
          links:
            - text: link3a
              url: file3a.md OR https://docs.microsoft.com/file3a
            - text: link3b
              url: file3b.md OR https://docs.microsoft.com/file3b
# footer (optional)
footer: "footertext [linktext](https://docs.microsoft.com/footerfile)"

```

Each section of the hub page template is described in more detail in the following sections. The `highlightedContent`, `productDirectory`, `conceptualContent`, `tools`, and `additionalContent` sections cannot be repeated and you cannot change the order. All of these sections are optional.

## Root

The root section populates the page title and the page summary. The `brand` key controls the color used in the hero and certain icons.



```

### YamlMime:Hub

title: service documentation # < 60 chars
summary: summary # < 160 chars
# brand: aspnet | azure | dotnet | dynamics | m365 | ms-graph | office | power-apps | power-automate |
power-bi | power-platform | power-virtual-agents | sql | sql-server | vs | visual-studio | windows | xamarin
brand: brand

```

## Metadata

The metadata section identifies the author of a hub page, is used for internal reporting, and includes text that is displayed in search results. In general, you should set the metadata for your hub page to what your team uses. Be sure to set the `ms.topic` key to `hub-page`. For the `description` key, follow the [SEO guidance](#). If you do not specify `title` and `description`, the metadata title and description will default to the title and summary provided in the root section. For more information, see [Metadata overview](#).



```

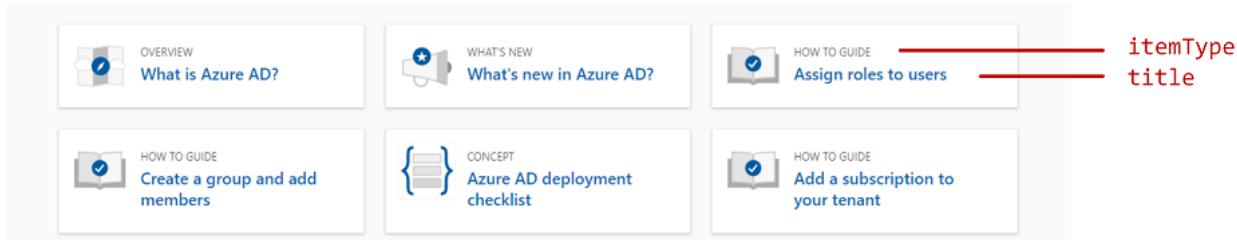
metadata:
  title: Azure Active Directory documentation
  description: Azure Active Directory (Azure AD) is Microsoft's multi-tenant, cloud-based directory, and identity management service that combines core directory services, application access management, and identity protection into a single solution.
  services: active-directory
  ms.service: active-directory
  ms.topic: hub-page
  ms.collection: M365-identity-device-management
  author: mtillman
  ms.author: mtillman
  ms.date: 01/27/2020

```

## highlightedContent

The `highlightedContent` section is designed for content that you want to highlight and is optional. This section can have up to 8 cards. Each card includes an image, an item type, and a title, and links to a single piece of content. The image and item type are based on the value of `itemType`.

ITEM	MIN	MAX
Number of cards	1	8



```

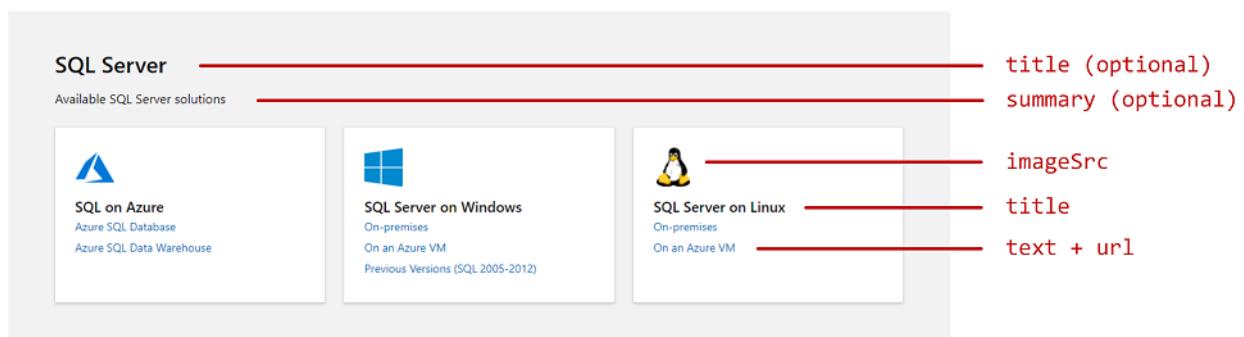
# highlightedContent section (optional)
# Maximum of 8 items
highlightedContent:
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
quickstart | reference | sample | tutorial | video | whats-new
  items:
    # Card
    - title: cardtitle1
      itemType: itemType
      url: file1.md OR https://docs.microsoft.com/file1
    # Card
    - title: cardtitle2
      itemType: itemType
      url: file2.md OR https://docs.microsoft.com/file2
    # Card
    - title: cardtitle3
      itemType: itemType
      url: file3.md OR https://docs.microsoft.com/file3

```

## productDirectory

The `productDirectory` section is designed for product-related content and is optional. This section can have a title and a summary, and can contain 25 cards. Each card includes an image, a title, and links. The image is one that you specify and should be square in ratio with no whitespace.

ITEM	MIN	MAX
Number of cards	1	25
Number of links on a card	1	6

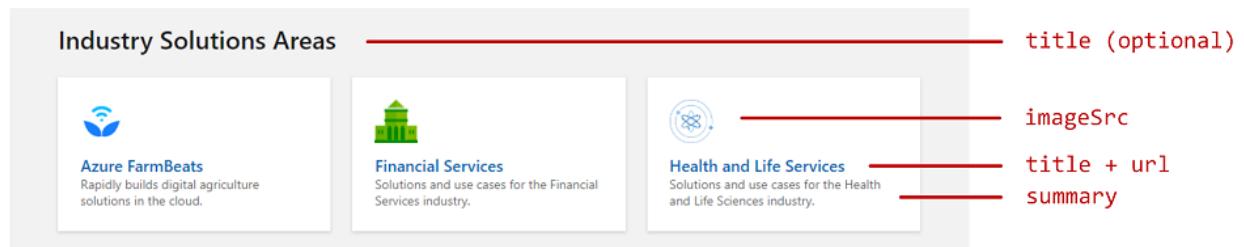


```

# productDirectory section (optional)
productDirectory:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      links:
        - url: file1a.md OR https://docs.microsoft.com/file1a
          text: linktext1a
        - url: file1b.md OR https://docs.microsoft.com/file1b
          text: linktext1b
    # Card
    - title: cardtitle2
      imageSrc: ./media/index/image2.svg OR https://docs.microsoft.com/media/logos/image2.svg
      links:
        - url: file2a.md OR https://docs.microsoft.com/file2a
          text: linktext2a
        - url: file2b.md OR https://docs.microsoft.com/file2b
          text: linktext2b
    # Card
    - title: cardtitle3
      imageSrc: ./media/index/image3.svg OR https://docs.microsoft.com/media/logos/image3.svg
      links:
        - url: file3a.md OR https://docs.microsoft.com/file3a
          text: linktext3a
        - url: file3b.md OR https://docs.microsoft.com/file3b
          text: linktext3b

```

The following shows the `productDirectory` section where each card has a summary. If you add a summary, you can only have a single link on the card. You can't have a card with multiple links and a also summary.



```

# productDirectory section (optional)
productDirectory:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      summary: cardsummary1
      url: file1.md OR https://docs.microsoft.com/file1
    # Card
    - title: cardtitle2
      imageSrc: ./media/index/image2.svg OR https://docs.microsoft.com/media/logos/image2.svg
      summary: cardsummary2
      url: file2.md OR https://docs.microsoft.com/file2
    # Card
    - title: cardtitle3
      imageSrc: ./media/index/image3.svg OR https://docs.microsoft.com/media/logos/image3.svg
      summary: cardsummary3
      url: file3.md OR https://docs.microsoft.com/file3

```

## conceptualContent

The `conceptualContent` section is designed for the bulk of your hub content and is optional. This section can have a title and a summary, and can contain 18 cards. The cards must have a title and at least one link. Links consist of the link type, text, and a URL. There is an option to add an additional footer link at the bottom of a card.

ITEM	MIN	MAX
Number of sections	0	3
Number of cards	2	18
Number of links on a card	1	12

The screenshot shows the conceptualContent section with three cards:

- Get started** (highlighted with a red box)
  - title (optional)**: "Get started"
  - summary (optional)**: "Use Visual Studio to edit, debug, and build code, and then publish an app."
- Learn how to use Visual Studio**
  - title**: "Learn how to use Visual Studio"
  - links**: A list of 8 items: Start a guided tour, Open code from a repo, Write and edit code, Build your code, Debug your code, Test your code, Access data locally or in the cloud.
- Follow a tutorial**
  - title**: "Follow a tutorial"
  - links**: A list of 7 items: C#, F#, Visual Basic, C++, Python, JavaScript.
- Create an app** (highlighted with a red box)
  - title**: "Create an app"
  - itemType**: "Universal Windows app" (highlighted with a red box)
  - links**: A list of 5 items: Windows desktop app, Universal Windows app, Mobile app, Unity game, Web app with ASP.NET Core.
  - text + url**: "See more >"
  - footerLink (optional)**: "See more >"
  - text + url**: "See more >"

```

# conceptualContent section (optional)
conceptualContent:
# Supports up to 3 sections
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview | quickstart | reference | sample | tutorial | video | whats-new
    title: sectiontitle # < 60 chars (optional)
    summary: sectionsummary # < 160 chars (optional)
    items:
        # Card
        - title: cardtitle1
            links:
                - url: file1.md OR https://docs.microsoft.com/file1
                    itemType: itemType
                    text: linktext1
                - url: file2.md OR https://docs.microsoft.com/file2
                    itemType: itemType
                    text: linktext2
                - url: file3.md OR https://docs.microsoft.com/file3
                    itemType: itemType
                    text: linktext3
        # footerLink (optional)
        footerLink:
            url: filefooter.md OR https://docs.microsoft.com/filefooter
            text: See more
        # Card
        - title: cardtitle2
            links:
                - url: file1.md OR https://docs.microsoft.com/file1
                    itemType: itemType
                    text: linktext1
                - url: file2.md OR https://docs.microsoft.com/file2
                    itemType: itemType
                    text: linktext2
                - url: file3.md OR https://docs.microsoft.com/file3
                    itemType: itemType
                    text: linktext3
        # footerLink (optional)
        footerLink:
            url: filefooter.md OR https://docs.microsoft.com/filefooter
            text: See more
        # Card
        - title: cardtitle3
            links:
                - url: file1.md OR https://docs.microsoft.com/file1
                    itemType: itemType
                    text: linktext1
                - url: file2.md OR https://docs.microsoft.com/file2
                    itemType: itemType
                    text: linktext2
                - url: file3.md OR https://docs.microsoft.com/file3
                    itemType: itemType
                    text: linktext3
        # footerLink (optional)
        footerLink:
            url: filefooter.md OR https://docs.microsoft.com/filefooter
            text: See more

```

## tools

The `tools` section is designed for tool, language, framework, CLI, or extension-type content that is associated or recognized with an image. This section is optional. This section can have a title and a summary, and can contain 12 cards. Each card includes of an image, a title, and the card URL. The image is one that you specify and should be square in ratio with no whitespace.

ITEM	MIN	MAX
Number of cards	2	16

Languages

Get started with a programming language.

C++	C#	F#	VB	Visual Basic
Python	JavaScript	TypeScript	R	

title (optional)  
summary (optional)

title  
imageSrc

```
# tools section (optional)
tools:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      url: file1.md
    # Card
    - title: cardtitle2
      imageSrc: ./media/index/image2.svg OR https://docs.microsoft.com/media/logos/image2.svg
      url: file2.md
    # Card
    - title: cardtitle3
      imageSrc: ./media/index/image3.svg OR https://docs.microsoft.com/media/logos/image3.svg
      url: file3.md
```

## additionalContent

The `additionalContent` section is designed for additional content that might go beyond your core hub content. This section is the most flexible and optional. This section can have up to 3 subsections that can include a title, a summary, 12 cards, and one footer subsection. There are two card styles: cards with a summary and cards with multiple links. Although you can use both styles at the same time, you should pick one style.

The following shows the `additionalContent` section for cards with the summary style. Each card includes a title, a summary (which allows for markdown), and a URL. The `url` key is only used if the `summary` key does not have a link. The `footer` section is the final optional subsection that is beneath the list of cards and allows for markdown.

ITEM	MIN	MAX
Number of sections	0	4
Number of cards per section	2	18
Number of links on a card	1	12

**Configuration Manager community & support**

Find related links to community and support.

The screenshot shows a grid of four cards:

- Configuration Manager blog**: News and announcements
- #ConfigMgr on Twitter**: Keep current with the active Twitter community
- Configuration Manager forums**: Ask technical questions in the product forums
- UserVoice product feedback**: Share product ideas with the engineering team

Below the cards is a link: "Found a problem on our site? [Let us know!](#)".

Annotations on the right side of the screenshot:

- title (optional)** (red line) points to the title of the first card.
- summary (optional)** (red line) points to the summary of the first card.
- title** (red line) points to the title of the third card.
- summary** (red line) points to the summary of the third card.
- footer (optional)** (red line) points to the "Let us know!" link.

```

# additionalContent section (optional)
# Card with summary style
additionalContent:
  # Supports up to 4 sections
  sections:
    - title: sectiontitle # < 60 chars (optional)
      summary: sectionsummary # < 160 chars (optional)
      items:
        # Card
        - title: cardtitle1
          summary: cardsummary1
          url: file1.md OR https://docs.microsoft.com/file1
        # Card
        - title: cardtitle2
          summary: cardsummary2
          url: file1.md OR https://docs.microsoft.com/file2
        # Card
        - title: cardtitle3
          summary: cardsummary3
          url: file1.md OR https://docs.microsoft.com/file3
  # footer (optional)
  footer: "footertext [linktext](https://docs.microsoft.com/footerfile)"

```

The following shows the `additionalContent` section for cards with the multiple links style. Each card includes a title and multiple links, and has a visual look similar to the `conceptualContent` section (minus the corresponding link icons). If you choose, you use the `note` key to append plain text to a link. The `footer` section is the final optional subsection that is beneath the list of cards and allows for markdown.

**Other content**

Other links related to Configuration Manager.

The screenshot shows a grid of three cards:

- Develop**: PowerShell, SDK concepts, SDK reference, SQL views
- Tools**: Support Center, MDT, Package Conversion Manager, See more >
- Other community sites**: ConfigMgr on Reddit, ConfigMgr Professionals Group on Facebook (Requires an account)

Below the cards is a link: "Found a problem on our site? [Let us know!](#)".

Annotations on the right side of the screenshot:

- title (optional)** (red line) points to the title of the first card.
- summary (optional)** (red line) points to the summary of the first card.
- title** (red line) points to the title of the third card.
- text + url** (red line) points to the text and URL of the second card.
- note (optional)** (red line) points to the note field of the third card.
- footer (optional)** (red line) points to the "Let us know!" link.

```

# additionalContent section (optional)
# Card with links style
additionalContent:
  # Supports up to 4 sections
  sections:
    - title: sectiontitle # < 60 chars (optional)
      summary: sectionsummary # < 160 chars (optional)
      items:
        # Card
        - title: cardtitle1
          links:
            - text: link1a
              url: file1a.md OR https://docs.microsoft.com/file1a
            - text: link1b
              url: file1b.md OR https://docs.microsoft.com/file1b
        # Card
        - title: cardtitle2
          links:
            - text: link2a
              url: file2a.md OR https://docs.microsoft.com/file2a
            - text: link2b
              url: file2b.md OR https://docs.microsoft.com/file2b
        # Card
        - title: cardtitle3
          links:
            - text: link3a
              url: file3a.md OR https://docs.microsoft.com/file3a
            - text: link3b
              url: file3b.md OR https://docs.microsoft.com/file3b
  # footer (optional)
  footer: "footertext [linktext](https://docs.microsoft.com/footerfile)"

```

## itemType

In the `highlightedContent` and `conceptualContent` sections, you specify the link type with the `itemType` key.

Links with the `itemType` modifier should link directly to doc content instead of another page with more links.

The value that you specify adds an icon for the link and also a category title for `highlightedContent`. For

`itemType`, you can use one of the following values. You cannot use any other values.

ICON	ITEMTYPE	DESCRIPTION
	architecture	Detailed guidance for how to design and deploy an architecture.
	concept	In-depth explanation of functionality related to a service(s) that are fundamental to understanding and use.
	deploy	How to deploy a service.
	download	Download a file related to the service or product.
	get-started	Getting started content that targets users that are new to a service or product.

ICON	ITEMTYPE	DESCRIPTION
	how-to-guide	Procedural articles that show the customer how to complete a task.
	learn	Typically Microsoft Learn modules, but can be any learning content.
	overview	Describes a product/service from a technical point of view.
	quickstart	Fundamental day 1 instructions that help new customers quickly use a product or service.
	reference	Documentation for APIs, PowerShell cmdlets, CLI commands, or other types of language-based content.
	sample	Related code sample.
	tutorial	Scenario-based procedures for the top customer tasks.
	video	Video that supplements the content.
	whats-new	Describes recent changes to product or service.

## Special characters

If you need to use special characters, enclose the entire string in double quotes ("").

SPECIAL CHARACTER	EXAMPLE
Colon (:)	- text: "Microsoft Ignite: Lock down access to Azure"
Number sign (#) # is a comment in YAML	- text: "See #azuredocs on Twitter"

## Linking to more information

You don't want to overwhelm customers with lots of links. For example, if your hub page has over 100 links, you should rethink your linking strategy. Instead, you should provide the top links that most customers will need. If you still have more content that you want to make sure is available, you can explore using more information links.

### See more links

Some customers will want additional details about a particular area. To support these customers, you can add See more links in the conceptualContent section. Use the footerLink key to link to more information such as a landing page or a relevant article. For an example of See more links, see the [Azure Active Directory documentation](#) hub page.

**Application management**

- ≡ [What is single sign-on \(SSO\)?](#)
- ≡ [Automatic user provisioning](#)
- ≡ [Application Proxy for on-premises apps](#)

[See more >](#)

## Search for more links

Depending on how your content is organized, you might not have a landing page or another appropriate page to link to for more information. If you really need to help customers discover more content, you can consider using a search query on docs.microsoft.com. If you use search query links, you should follow these guidelines:

- Use **Search for more** for the link text.
- Construct a search query that returns relevant results that can help the customer.
- Use Docs search (<https://docs.microsoft.com/search/>) instead of Azure search (<https://azure.microsoft.com/search/>).
- Define search scopes by using the `searchScope` property.
- Your search query should have a format similar to the following:
  - <https://docs.microsoft.com/search/?search=your+string&scope=yourscope&category=category>

In docfx.json, you can define a folder-level search scope. In the following example, an Azure Active Directory (AAD) scope is defined for all content under docs.microsoft.com/azure/active-directory/.

```
"fileMetadata": {  
    "searchScope": {  
        "articles/active-directory/**/*": ["AAD", "Azure"]  
    },
```

If you want to list all Azure Active Directory content that includes the "Conditional Access" string, your query would look like the following:

- <https://docs.microsoft.com/search/?search=conditional+access&scope=AAD&category=All>

If you just want to list conceptual content, you would set the category parameter to `Documentation`:

- <https://docs.microsoft.com/search/?search=conditional+access&scope=AAD&category=Documentation>

Validate the results of your search query and adjust the search string or scope as needed. For more information, see [Docs Search](#) in the Onboarding Guide.

## Required criteria

When creating your hub page, you must follow this criteria:

- Title must be sentence cased. End title with **documentation**.
- Maximum of 3 section types. For example, your hub page could use the `highlightedContent`, `conceptualContent`, and `additionalContent` sections, but no more section types.
- Requires approval from the designated [business approver](#).

## Guidelines

When creating your hub page, you should follow these guidelines:

- Cards and links should be based on top customer tasks, top products, or top services.
- Except for **See more** links, all links should link directly to doc content instead of another page with more links.
- Don't duplicate links.
- When possible, use **relative paths** for links.
- For card titles, try to use tasks or simple language instead of product names.
- Avoid using both card styles in `additionalContent`. Pick either the card-with-summary style or the card-with-links style.
- If you have a `conceptualContent` card with a `footerLink`, use **See more** for the link text.
- If you have a **search query link**, use **Search for more** for the link text.
- For the `productDirectory` and `tools` sections, use images that are square in ratio with no whitespace.

## Choose the right content for your hub page

The categories and specific content in each section of the hub page should be focused on answering these questions:

### 1. What is this product?

Users may be encountering the product documentation for the first time and trying to get their bearings. Other users may be very familiar with the product, but want to know if there are any new features or bugs they should be aware of. Content such as overviews or *What's New* articles should be given priority on the page, such as within the first set of cards in the `highlightedContent` section.

### 2. What can I do with it?

Both new and seasoned users want to know the types of tasks that can be accomplished or supported by using a specific product. Having task-focused categories and content is a great way to orient users and help them quickly find the information they need to solve their issue or learn more about specific features.

### 3. How can I get started?

Links to Learn modules or other training resources about using the product or getting a product up and running for a new user are extremely valuable to people looking at documentation.

### 4. How does this product work alongside other Microsoft products?

Links to content about how a product or service interacts with other Microsoft products is beneficial to users as it gives them more context for what a product is and does. Including this content on a hub page also helps showcase the breadth of Microsoft products.

Product teams can help you understand the possible answers to these questions as well as help you identify the likely top tasks and priority information based on customer feedback.

## Information architecture tips

- Do group similar content as much as possible.

Task-based content should be grouped alongside other task-based content, Learn and other educational content is likely to be grouped together—whatever your main organizing categories are, make sure that like appears with like.

- Don't create *Other* and *Miscellaneous* categories.

These types of categories are often a catch-all for content that doesn't really fit with the rest of the content promoted on the page. While it's tempting to throw content into an *other* category, this isn't a useful label or understandable to people. If you find yourself unsure of how to fit certain content on the page, first revisit your organizing categories—there may be an easy way to expand a category to fit a

piece of important content. If there are no established categories for that content to fit, also question whether a link to that specific content is actually needed on the hub page.

- Do include section headers or descriptive copy on cards.

This information can help users scan the page and quickly find links relevant to what they are looking for, providing more context for why a link is being included on the hub page.

- Don't include too many links to external sites.

The majority of the content linked to from the hub page should be within the Docs environment. Linking off to external sites or other sections of the larger Microsoft domain can be disruptive to a user's task and may end up confusing them more than helping. If you are linking off to an external site, make that clear either in link labels, additional microcopy, or iconography on the page.

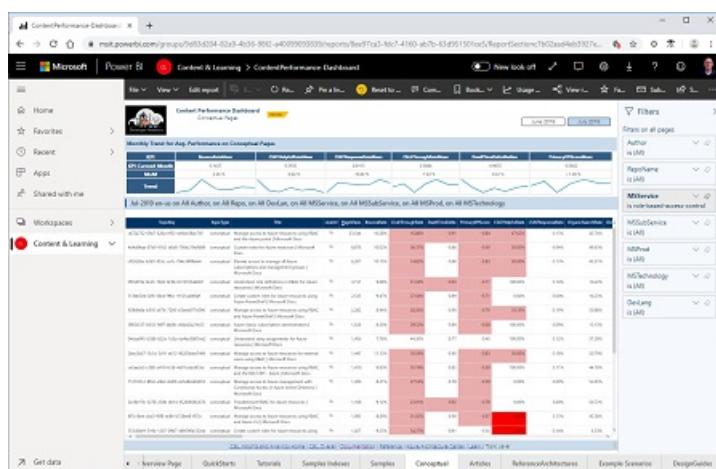
- Do incorporate user research for complex content sets.

If you have a complex set of content that even you're not sure how to categorize, this is a great area to do some light user research to understand the categories that your users understand. This could be a simple card sort where you have participants sort specific content into their own groupings, or more in-depth research interviews to understand the mental models people have around a topic or type of product.

If you have questions about the best ways to organize content on your hub page or want to know more about the ways you can use user research to discover the most useful organization for your audience, please [contact the DevRel Information Architecture team](#).

## Steps to create or update your hub page

1. Determine all docsets that should be represented on your hub page.
  2. Work with your product team to identify the top customer tasks, top products, or top services for your hub content. These will help determine which cards to have.
  3. Review the [Content Performance dashboard](#) metrics for your hub content. If you are updating an existing hub page, review the [Microsoft Docs Metrics tool's](#) click numbers. These metrics will help determine which links to add to your cards.



4. Review the [required criteria](#) and [guidelines](#).
  5. Start with the [hub page template](#) to create your cards and links.
  6. Create or update your `index.yml` file in the root folder of your hub content.
  7. If you have an existing `index.md` file, delete that file.
  8. If your existing index file used images that are your own and not shared with other pages, delete those images.

images.

9. Create a pull request.
10. Continue to iterate until you have a pull request without any errors.

Alternatively, you can have the [Developer Relations Content Production Service \(CPS\) team](#) create a pull request. If you want to have the CPS team create a pull request, follow these steps:

1. Copy the [hub page template](#) into a Word document.
2. Update the Word document with your text and links.
3. Follow the steps at [Hub/Landing Page, TOC Conversion, and UHF Services](#) to create a work request.
4. Attach the Word document to the work request.

The CPS team will send you a link to the pull request.

## Get pull request reviewed and approved

You should budget some time to get your hub page reviewed and approved. If you are converting your hub page to the new design, you will likely need to make updates as part of the review and approval process.

1. (Optional) Send your pull request to [docshub@microsoft.com](mailto:docshub@microsoft.com) to get early feedback.
2. Once you have a pull request without any errors, #sign-off on the pull request.
3. If your repo has pull request reviewers, they will review your changes.
4. If your pull request passes the [quality criteria](#), the pull request reviewer will contact the designated [business approver](#) to request approval.

Hub pages require approval from the business approver.

5. If approvers have any changes, you must make the specified changes and #sign-off again.
6. Once your pull request has been approved by two approvers, the pull request reviewer can merge your changes.

## Analyze your page

Once you publish your hub page, how do you determine how it is performing? Here are a couple steps you can follow.

1. Use the [MDM Edge and Chrome extension](#) to look at the clicks on the page.

On pages designed for navigation, like hubs and landing pages, it's useful to identify links lower on the page that get more clicks. Move them up.

Note that MDM has a limitation in that it cannot differentiate between the same links on a page. Both links will display the same number of clicks.

2. Open the [Content Performance dashboard](#), find your hub page, and look at the [BounceRate](#), [ClickThroughRate](#), and [ExitRate](#). If these metrics are displayed in red or pink, follow the steps in [How to troubleshoot lower-performing articles](#) to improve the metrics.

## Troubleshoot

When you create your hub page, you might encounter errors. This table describes some issues and the corresponding solution.

EXAMPLE ERROR MESSAGE	SOLUTION
Array item count 9 exceeds maximum count of 8. Path 'highlightedContent.items'.	Reduce the number of cards in the highlightedContent section to 8 or under.
Required properties are missing from object: imageSrc. Path 'productDirectory.items[0]'.	In the productDirectory section, ensure the imageSrc path is correct or add the missing image.
Required properties are missing from object: itemType. Path 'conceptualContent.items[2].links[3]'. Invalid type. Expected String but got Null. Path 'conceptualContent.items[2].links[4].itemType'.	In the conceptualContent section, use a valid <code>itemType</code> .
Invalid type. Expected Array but got Object. Path 'additionalContent.sections'.	Ensure that the YAML in the specified section matches the format in the template.

## Questions

If you have questions or problems related to hub pages, send email to:

- [docshub@microsoft.com](mailto:docshub@microsoft.com)

## Related links

- [Hub page schema](#)
- [Hub page spec](#)
- [Hub & Landing Page Pilot Analysis](#)
- [Update Azure docs hub page](#)
- [DevRel LandingPages tracking dashboard](#)

# Create or update a landing page

6/16/2021 • 8 minutes to read

This topic is for Azure services and other products that follow the [Content & Learning content model](#). It describes the high-level process to create or update a landing page. To watch a short overview video of the process, see [Video: How to update a landing page](#).

<https://aka.ms/docslanding>

## Overview

Landing pages are an entry point for your customers into a single docset. Landing pages typically map to a single TOC and appear at the top of the TOC. You use landing pages to surface the top customer tasks or top subjects. Links on a landing page should primarily link directly to articles within your docset. A landing page could be a child of an overall [hub page](#) strategy. The following diagram shows how you might use a landing page in your content structure.



## Design

Starting in August 2019, a card-based landing page design was implemented that replaces the previous design. All teams in Content & Learning should use this new design going forward. This new design has the following benefits:

- Improve browsing experience for users to find content faster by creating a more dense presentation of info and revealing more content
- More focused on top customer tasks, products, or services
- Create consistent user experiences for hub and landing pages
- Support products of varying complexity and depth
- Simpler design

Here's an example of the **new** landing page design:

The screenshot shows the Microsoft Azure RBAC documentation landing page. At the top, there's a navigation bar with links like Overview, Solutions, Products, Documentation, Pricing, Training, Marketplace, Partners, Support, Blog, More, and a Free account button. Below the navigation is a breadcrumb trail: Azure / Role-based access control. On the left, there's a sidebar with a 'Filter by title' dropdown and a list of topics under 'RBAC for Azure resources documentation'. The main content area has three columns: 'About RBAC for Azure resources' (with sections for Overview, What is RBAC?, Understand the different roles, and Microsoft Ignite: Lock down access to Azure), 'Get started' (with sections for Quickstart, View access for a user, Tutorial, Grant a user access - Portal, Grant a group access - PowerShell, and Learn), and 'Manage access' (with sections for How to guide, Portal, PowerShell, Azure CLI, REST API, and Template). At the bottom left is a 'Download PDF' link.

Here's an example of the previous landing page design:

The screenshot shows the previous design of the Microsoft Azure RBAC documentation landing page. The layout is similar to the new one, with a navigation bar at the top, a breadcrumb trail, and a sidebar on the left. However, the main content area is organized differently. It features a large section for 'RBAC for Azure resources documentation' with a detailed description of what RBAC is and how it manages access. To the right of this, there's a video thumbnail for 'Lock down access to Azure (1:16)' featuring Stuart Klein. Below the main description are sections for '5-minute quickstarts' (with a 'View access for a user' link) and 'Step-by-step tutorials' (listing 'Grant a user access - Portal', 'Grant a group access - PowerShell', 'Create a custom role - PowerShell', and 'Create a custom role - CLI'). Further down is a 'Build your skills with Microsoft Learn' section and a 'Secure your Azure resources with role-based access control (RBAC)' link. A feedback pop-up in the bottom right corner asks 'Is this page helpful?' with 'Yes' and 'No' buttons.

## Example landing pages

LANDING PAGE	SOURCE
Azure Active Directory Identity Governance documentation	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory/governance/index.yml">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory/governance/index.yml</a>
Azure Data Box documentation	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/databox/index.yml">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/databox/index.yml</a>
Azure AD Domain Services documentation	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory-domain-services/index.yml">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory-domain-services/index.yml</a>
Azure IoT Edge documentation	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/iot-edge/index.yml">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/iot-edge/index.yml</a>
Microsoft identity platform	<a href="https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory/develop/index.yml">https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory/develop/index.yml</a>

## Landing page template

To create your landing page, you use YAML. The source file is an index.yml file in the root folder of the docset. The following template defines the structure. Add this structure to the index.yml file.

**metadata:title**

**title**

**summary**

**linkListType**

**text + url**

**linkList**

```
#### YamlMime:Landing

title: service documentation # < 60 chars
summary: summary # < 160 chars

metadata:
    title: title # Required; page title displayed in search results. Include the brand. < 60 chars.
    description: description # Required; article description that is displayed in search results. < 160 chars.
    ms.service: service #Required; service per approved list. service slug assigned to your service by ACOM.
    ms.subservice: subservice # Optional; Remove if no subservice is used.
```

```
ms.topic: landing-page # Required
ms.collection: collection # Optional; Remove if no collection is used.
author: githubauthor #Required; your GitHub user alias, with correct capitalization.
ms.author: msauthor #Required; microsoft alias of author; optional team alias.
ms.date: 01/27/2020 #Required; mm/dd/yyyy format.

# linkListType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
quickstart | reference | sample | tutorial | video | whats-new

landingContent:
# Cards and links should be based on top customer tasks or top subjects
# Start card title with a verb
# Card (optional)
- title: About [service]
linkLists:
- linkListType: overview
links:
- text: What is [service]?
url: file1.md
- text: linktext2
url: file2.md
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md

# Card (optional)
- title: Get started
linkLists:
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md

# Card
- title: cardtitle3
linkLists:
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md

# Card
- title: cardtitle4
linkLists:
- linkListType: linkListType
links:
- text: linktext1
url: file1.md
- text: linktext2
url: file2.md
```

```

- linkListType: linkListType
  links:
    - text: linktext1
      url: file1.md
    - text: linktext2
      url: file2.md

```

## Metadata

The metadata section identifies the author of the landing page, is used for internal reporting, and includes text that is displayed in search results. In general, you should set the metadata for your landing page to what your team uses. Be sure to set the `ms.topic` key to `landing-page`. For the `description` key, follow the [SEO guidance](#). If you do not specify `title` and `description`, the metadata title and description will default to the title and summary provided in the root section. For more information, see [Metadata overview](#).

```

metadata:
  title: RBAC for Azure resources documentation
  description: Learn about role-based access control (RBAC) to manage who has access to Azure resources, what they can do with those resources, and what areas they have access to.
  services: active-directory
  ms.service: role-based-access-control
  ms.topic: landing-page
  author: rolyon
  ms.author: rolyon
  ms.date: 01/27/2020

```

[Role-based access control \(RBAC\) for Azure resources documentation](https://docs.microsoft.com/en-us/azure/role-based-access-control/) DOCUMENTATION ————— **metadata:title**  
<https://docs.microsoft.com/en-us/azure/role-based-access-control/>

Learn about role-based access control (**RBAC**) to manage who has access to Azure resources, what they can do with those resources, and what areas they have access to ————— **metadata:description**

## linkListType

Links on a card are grouped into categories that you specify with the `linkListType` key. For `linkListType`, you can use one of the following values. You cannot use any other values.

ICON	LINKLISTTYPE	DESCRIPTION
	architecture	Detailed guidance for how to design and deploy an architecture.
	concept	In-depth explanation of functionality related to a service(s) that are fundamental to understanding and use.
	deploy	How to deploy a service.
	download	Download a file related to the service or product.
	get-started	Getting started content that targets users that are new to a service or product.

ICON	LINKLISTTYPE	DESCRIPTION
	how-to-guide	Procedural articles that show the customer how to complete a task.
	learn	Typically Microsoft Learn modules, but can be any learning content.
	overview	Describes a product/service from a technical point of view.
	quickstart	Fundamental day 1 instructions that help new customers quickly use a product or service.
	reference	Documentation for APIs, PowerShell cmdlets, CLI commands, or other types of language-based content.
	sample	Related code sample.
	tutorial	Scenario-based procedures for the top customer tasks.
	video	Video that supplements the content.
	whats-new	Describes recent changes to product or service.

## Limits

The schema for a landing page has built-in limits. If you exceed these limits, you will get a validation error when building.

ITEM	MAX
Number of cards	12
Number of <code>linkListType</code> in a card	6
Number of links in a <code>linkListType</code>	10
Custom icons or images	Not allowed

## Special characters

If you need to use special characters, enclose the entire string in double quotes (`" "`).

SPECIAL CHARACTER	EXAMPLE
Colon (:)	<code>- text: "Microsoft Ignite: Lock down access to Azure"</code>

SPECIAL CHARACTER	EXAMPLE
Number sign (#) # is a comment in YAML	- text: "See #azuredocs on Twitter"

## Required criteria

When creating your landing page, you must follow this criteria:

- Minimum of 3 cards and a maximum of 12.
- Title must be sentence cased. End title with **documentation**.

## Guidelines

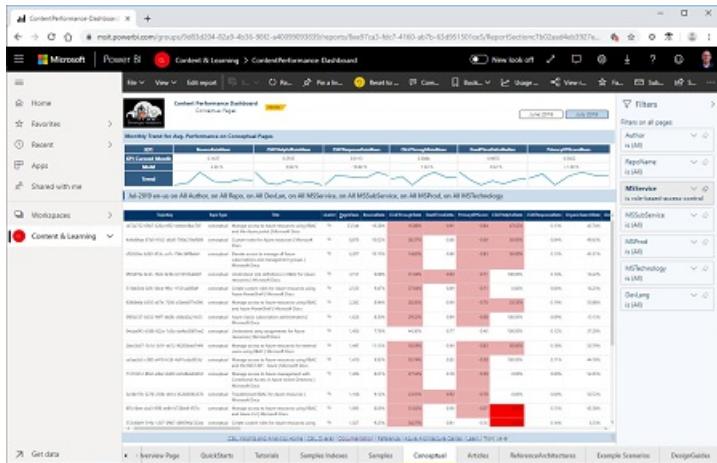
When creating your landing page, you should follow these guidelines:

- Cards and links should be based on top customer tasks or top subjects.
- Don't just repeat the TOC.
- Order cards by relevancy.
- Card titles should start with a verb.
- When possible, use [relative paths](#) for links.
- `linkListType` should match the [content type](#).
- Don't duplicate links.
- Link directly to doc content instead of a hub page, landing page, or another page with more links.
- Number of links outside your docset should be **4** or less.
  - Other docs on doc.microsoft.com
  - Articles on other Microsoft properties
  - External sites
  - Learn modules
  - Videos
- Reference links are **not** considered outside your docset.
  - PowerShell cmdlets
  - CLI commands
  - APIs and SDKs

## Steps to create or update your landing page

This section describes the steps to create or update your landing page. To watch a short overview video of the process, see [Video: How to update a landing page](#).

1. Work with your product team to identify the top customer tasks or top subjects for your docset. These will help determine which cards to have.
2. Review the [Content Performance dashboard](#) metrics for your docset. If you are updating an existing landing page, review the [Microsoft Docs Metrics tool's](#) click numbers. These metrics will help determine which links to add to your cards.



3. Review the [required criteria and guidelines](#).
  4. Start with the [landing page template](#) to create your cards and links.
  5. Create or update your `index.yml` file in the root folder of your docset.
  6. If you have an existing `index.md` file, delete that file and update your `TOC.yml` file.
  7. If your existing index file used images that are your own and not shared with other pages, delete those images.
  8. Create a pull request.
  9. Continue to iterate until you have a pull request without any errors.

Alternatively, you can have the [Developer Relations Content Production Service \(CPS\) team](#) create a pull request. If you want to have the CPS team create a pull request, follow these steps:

1. Copy the [landing page template](#) into a Word document.
  2. Update the Word document with your text and links.
  3. Follow the steps at [Hub/Landing Page, TOC Conversion, and UHF Services](#) to create a work request.
  4. Attach the Word document to the work request.

The CPS team will send you a link to the pull request.

# Get pull request reviewed

1. Once you have a pull request without any errors, #sign-off on the pull request.
  2. If your repo has pull request reviewers, they will review your changes.
  3. If your pull request passes the [quality criteria](#), the pull request reviewer can merge your changes.

Landing pages no longer require business approval.

## Analyze your page

Once you publish your landing page, how do you determine how it is performing? Here are a couple steps you can follow.

1. Use the [MDM Edge and Chrome extension](#) to look at the clicks on the page.

On pages designed for navigation, like hubs and landing pages, it's useful to identify links lower on the page that get more clicks. Move them up.

Note that MDM has a limitation in that it cannot differentiate between the same links on a page. This also includes a link on a page and the same link in the TOC. Both links will display the same number of clicks.

2. Open the [Content Performance dashboard](#), find your landing page, and look at the [BounceRate](#), [ClickThroughRate](#), and [ExitRate](#). If these metrics are displayed in red or pink, follow the steps in [How to troubleshoot lower-performing articles](#) to improve the metrics.

## Troubleshoot

When you create your landing page, you might encounter errors. This table describes some issues and the corresponding solution.

EXAMPLE ERROR MESSAGE	SOLUTION
Array item count 13 exceeds maximum count of 12. Path 'landingContent'.	Reduce the number of cards to 12 or under.
Array item count 11 exceeds maximum count of 6. Path 'landingContent[0].linkLists'.	Reduce the number of linkListTypes in a card to 6 or under.
Array item count 25 exceeds maximum count of 10. Path 'landingContent[0].linkLists[0].links'.	Reduce the number of links in a linkListType to 10 or under.
Value "custom" is not defined in enum. Path 'landingContent[0].linkLists[2].linkListType'.	Use a valid <a href="#">linkListType</a> .

## Videos

## Questions

If you have questions or problems related to landing pages, send email to:

- docshub@microsoft.com

## Related links

- [Landing page schema](#)
- [Landing page spec](#)
- [Hub & Landing Page Pilot Analysis](#)
- [DevRel LandingPages tracking dashboard](#)

# Information architecture on Docs

5/12/2021 • 3 minutes to read

This article introduces the core information architecture (IA) components used to structure content on docs.microsoft.com.

## Breadcrumbs

Breadcrumbs are a series of links at the top of a page, which convey its position in the overall site hierarchy. They allow a user to navigate through the hierarchy, back to the homepage, one level at a time by starting at the last link in the breadcrumb trail. They are a fundamental way for us to teach users a mental model for how Docs is structured, allowing them to move around Docs more easily.

Resources: Review the [Breadcrumb guidelines](#) to understand how your breadcrumbs should be structured.

## Header Navigation

The Docs navigation header is the navigation bar at the top of every Docs page. It is made of a global header (the same on every page) and an L2 header (specific to an area of Docs). The IA team stewards header navigation in order to ensure that it establishes Docs as a place with clear boundaries, that it gives users a consistent view of the neighborhoods within docs.microsoft.com, and that it provides a reliable method for navigating between them.

Resources:

- Review [product-level header guidelines](#) to learn more about how to set up and structure your product-level header. Individual content teams maintain product-level headers, with partnership from IA and automated validation to ensure compliance with established standards.
- Get some background on [how we developed these navigation standards](#).

## Hub and landing pages

Hub and landing pages are navigational pages intended to help the user understand the structure of the content set they are consuming, as well as how to move through that set. Along with TOCs, they teach the user a mental model for how any given area of documentation is organized.

Resources: Review the [hub pages guide](#) and the [landing pages guide](#) for details and guidance on selecting and structuring your hub and landing pages.

## Metadata and taxonomies

We use metadata on Docs for reporting, discoverability of the content via search, and to drive aspects of the site experience. Certain metadata are user-facing, meaning that they directly appear somewhere in the Docs interface for customers to interact with, primarily in browse pages (such as Learn browse). The IA team governs the taxonomies for user-facing metadata.

Resources:

- See the [user-facing Docs taxonomies](#).
- Learn more about [Docs metadata](#) and how to apply it or request changes to it.

## Table of contents (TOC)

The table of contents appears on the left side of the page on docs.microsoft.com. It is limited to documentation content pages – TOCs do not appear in Learn, Code Samples, or Q&A. TOCs are the only real affordance we provide for users to navigate through docsets. It's important that they are usable, logical, and consistent in structure and behavior.

Resources: Review [TOC structure](#) for detailed guidelines on how to structure your TOC for the best customer experience.

## Unified content model (UCM)

The unified content model is an effort to ensure the content on Docs.microsoft.com is structured in a way that supports the business and customers. Structured content is more sustainable, easier to reuse, more efficient to manage, and supports sophisticated technical and customer experiences. The unified content model will make it easier for customers to understand and use our content, will make it easier for us to deploy and manage content, and will make it easier for us to build innovative experiences.

Resources:

- Review the [unified content model](#) in detail.
- Learn more about the [history of the unified content model and where it's headed](#).

## URLs

A strict file naming convention helps to clearly identify articles and contributes towards discoverability on the web. For most content, file and folder names are included in the public URL on docs.microsoft.com, so choose them carefully. The IA team is responsible for approving new base URLs.

Resources: Review [File name and path guidelines](#) to learn how to create good URLs.

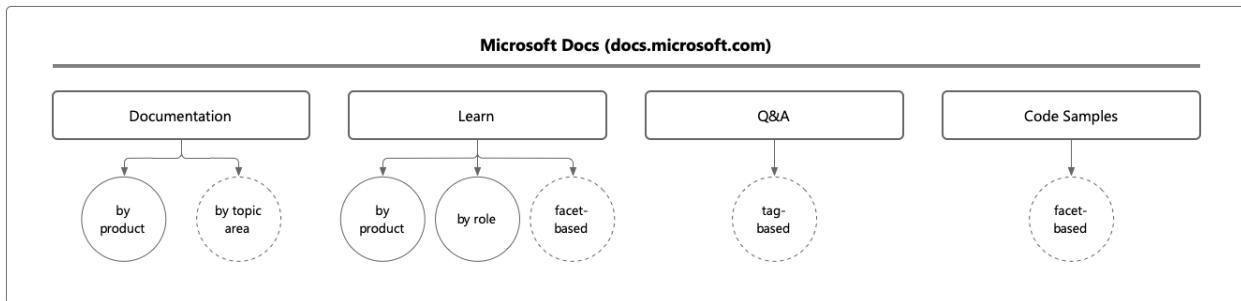
# Header navigation governance

11/2/2020 • 7 minutes to read

## Background

The header navigation on Microsoft Docs (the platform as a whole) consists of multiple layers:

- A global header that is consistent across the entire site. It is managed by IA + Docs product PM.
- Site-level headers ("L2 Navigation") that are specific to each neighborhood of Docs:
  - Product-specific headers within documentation: managed by individual content teams, with IA + Docs PM as consultants/governance managers, and automated validation
  - Learn header navigation: managed by IA + Learn product PM
  - Q&A header navigation: managed by IA + Q&A product PM
  - Code Samples: doesn't use a header, but depends on a faceted browse experience for navigation.  
Taxonomies managed by IA.



## Global navigation governance

### What goes in the global Docs header

The global level of Docs navigation provides access to the top-level subsites of Docs. Subsites are defined as "meaningful neighborhoods of Docs". A meaningful neighborhood has identifying characteristics:

1. It has a distinct, branded Microsoft identity (such as "Microsoft Learn" or "Microsoft Q&A"). It has gone through a naming convention review.
2. It contains content about more than one product. Note: To keep the global navigation lightweight enough to use, we don't include product identities in this definition because there are over 800 we need to support – see 'Documentation navigation' to learn about how we handle products in navigation.
3. It's oriented around a particular type or modality of content (such as guided learning (Learn) versus reference materials (Documentation) versus user-generated questions (Q&A)).
4. It's a sizeable collection of content. More than a few hundred pages.
5. It has a long lifespan. It's not a fleeting collection of items; it is intended to be a primary pillar of the Microsoft Docs experience.
6. It isn't a child of a subsite. Only top-level subsites are shown in the global header navigation.

To date, the business has distinguished these neighborhoods as: Documentation, Learn, Q&A, and Code Samples.

The global header also includes globally available tools and utilities that apply across all of Docs, which are: search and profile.

### How we assess global navigation requests

The global header sets fundamental user expectations for the architecture of Docs.microsoft.com as a whole. Any requests for additions to the global header are considered major and warrant a full review by an IA + Docs product PM.

In short, only subsites and global tools are candidates for the global header.

#### **Exceptions**

If an exception to the above decision is being considered, take the following into consideration:

1. Decisions that do not reflect the established site architecture will make it more difficult for users to develop a mental model of Docs, making it harder for them to navigate anywhere intentionally.
2. Decisions made now will be used as a precedent later. For example, adding a product-specific item to the global header now will make it more difficult to say no to a similar request in the future.

## Learn navigation governance

### **What goes in the Learn header**

The Learn navigation header provides access to the top-level meaningful areas of Learn. Today, the Learn header uses a content-type mechanism defining "meaningful areas":

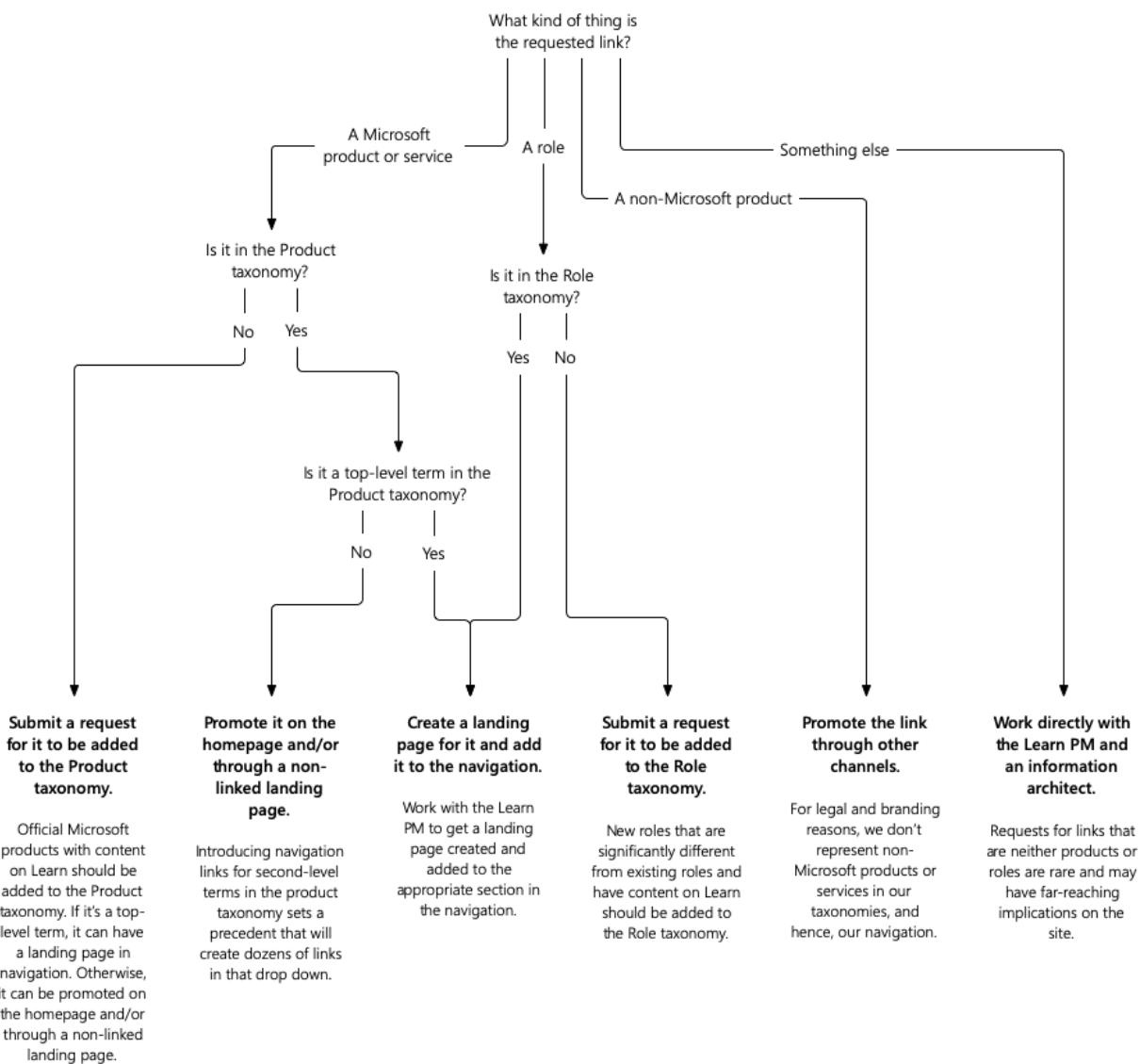
- Learning paths & modules (organized by Product and by Role)
- Learn TV (videos)
- Certifications & courses

Navigation and taxonomy are the two fundamental components of a site's information architecture. Well-aligned navigation and taxonomy make a site intelligible and usable for customers. Navigation and taxonomy that present inconsistent perspectives on the domain make a site harder to understand and use.

For this reason, we align top navigation and the taxonomies we use on the browse pages and on content. **Any large-scale organizational mechanisms on the site (such as organizing objects in navigation by product or by role) must be aligned with their relevant taxonomies.**

### **How we assess navigation requests**

New requests for links in the Learn navigation can be triaged using key questions as illustrated in this decision tree diagram:



In short, new product or role landing pages should go in the navigation if they are currently or are good candidates to be [top level taxonomy terms](#). If they are not, they should be promoted in other ways. Non-Microsoft products should never go in the navigation or taxonomy.

Taxonomy requests should be submitted to [DevRellA@service.microsoft.com](mailto:DevRellA@service.microsoft.com)

### Exceptions to Learn navigation requests

If an exception to the above decision tree is being considered, take the following into consideration:

1. There may be legal or branding implications to representing third-party products as Microsoft products or changing established names. Consider getting a review from CELA on any major exceptions. Most will be fine, but it's a wise standard precaution.
2. Moving a term in the taxonomy hierarchy will change how its content appears on the site. This is a technical limitation and cannot be changed. For example, in "Azure > Azure Active Directory", all of Azure Active Directory's content will appear under Azure. If Azure Active directory becomes a top-level term next to Azure, its content will no longer appear under Azure.
3. Multiple backend services access our taxonomies for many purposes. They have nuanced structural requirements and can't be arbitrarily changed in one place without affecting some other part of the site.
4. Our product taxonomy is very large. Approximately 200 Microsoft products are represented on Learn and 600 more are elsewhere on Docs.Microsoft.com. Because of this scale, we have to consider what would happen if this decision was made for every equivalent product. Many decisions that will work with eight products will not work with 800.
5. Decisions made now will be used as a precedent later. For example, adding a landing page to the navigation

for a second-level product term now will make it more difficult to say no to a similar request in the future.

## Q&A navigation governance

The Q&A navigation header provides access to the primary object types in Q&A. This particular navigation approach is nascent and still maturing. Today it uses an object-oriented approach, in which each item in the navigation is a unique type of object in the Q&A experience:

- Questions
- Tags
- Users
- FAQ & Help materials

We don't surface product or role lists in Q&A navigation today. If we ever do, they must align with the product & role taxonomies in the same manner as Learn navigation does.

Requests for additions to Q&A navigation are rare. Because the Q&A navigation is still developing, we need to take care when determining what to include or exclude, as it should trigger a reassessment of the entire navigation approach for Q&A. Any requests for additional links in navigation should be reviewed by IA and the Q&A PM.

## Documentation navigation governance

Documentation L2 headers are specific to the product area the user is in. Each product area gets its own L2 header. This header is designed to provide access to the meaningful areas of that product's documentation. The documentation neighborhood will emphasize site search and a product-directory on its homepage as its primary navigational affordances. Once a user navigates to specific product content, they will see a product-specific L2 header (such as [Azure](#) or [Visual Studio](#)).

A product's L2 navigation is managed by content owners. Its structure and implementation is governed by IA and Docs PM, through [product navigation guidelines](#) and automated validation. The IA team does not manage requests for changes to individual product L2 navigation; requests should be directed to the appropriate content owners.

## Navigation design principles

The [Unified Navigation v-team](#), with representatives from across DevRel and Azure, established a set of navigation principles for our technical content ecosystem. These principles state that navigation should:

- **Prioritize the customer experience over the org chart.** We don't need to reflect arbitrary team boundaries in navigation, as this doesn't help users understand our content or find it.
- **Favor clarity.** we don't bring superficial complexity to the experience. Our world of products is already complex; we don't need to add more cognitive load for users to sort through.
- **Be inclusive for all personas and archetypes.** We will not favor a particular role or audience in broad navigation work. We will honor the Microsoft principle of creating inclusive experiences.
- **Be predictable.** No surprises to the user and rely on common mental models.
- **Be durable across spaces.** Flexible in time to allow for adjustments and change in the future.
- **Be complementary to the user, content, and business.** We do not prioritize one thing (such as conversions) over informing the user "Where am I, what is here, and where can I go?"
- **Act as a system.** Navigation will honor the One Microsoft principle. No single site is more special or unique than any other. We don't prioritize exceptions.

# Content release planning + process

5/10/2021 • 8 minutes to read

Most of the time, authors and content teams are expected to manage their own releases. However, not all releases are the same! When the scope of a publishing event gets large enough, the content release is centrally managed. Currently, for the Content & Learning content portfolio, the Build and Ignite conferences are centrally managed for Azure technical content.

## Request a pre-review for a pull request containing a handful of articles you need published on a specific date

For a single PR that is for any significant and/or timed release, you can request a "pre-review" from the PR review team 2-3 business days before your release. A pre-review will allow you to get feedback on the pull request **before** you sign off. The turnaround for pre-reviews can take longer. Signed-off PRs are higher priority. Most of the time, they're addressed in less than 12 hours. When requesting a pre-review, make sure you're not planning any other changes (that is, it's ready for merge). Also, to prevent accidental early merge, DO NOT sign off in the PR. For repos where the Content & Learning vendor PR review team reviews pull requests, contact the [techdocprs](#) alias with your pre-review request.

When you have more than a small amount of new content, use a release branch. Even if only one author is working on the release it's best to use a release branch. And, you might start small in your fork and the scope could increase beyond five new articles. If that happens, request a release branch and start submitting pull requests against the release branch.

When you're managing your content release, you'll need to consider [the publishing schedule for your repo](#).

## Release planning process for self-managed releases with a release branch (general process)

### NOTE

If you're using `azure-docs-pr`, make sure to check out the next section also.

1. Request a release branch from the admin of your repository.
2. Create a release plan and share it with the authors who are collaborating on the release. See the release plan template later in this article. It has details on the minimum elements a release plan.
3. Work with your repo admin to plan the merge to the master and live branches.

## Release planning process for self-managed releases with a release branch (`azure-docs-pr`)

The Azure documentation repo is large and busy. Always file a ticket with the Content Production Service team to have release branches created and to schedule release branch merging to master.

### Release branches

To request a release branch, [file a ticket with the Content & Learning content production team](#). If you don't have perms to file a ticket, request perms at <http://myaccess/>. You need access to this Azure DevOps repo:

<https://ceapex.visualstudio.com/cps/>. More detailed instructions are available on the content production team site.

## Pull requests to release branches

Always submit new content to your release branch in small batches that meet the pull request criteria. Submitting content in small batches lets the PR review team review it iteratively. If you submit more than five new articles in the same pull request, the PR review team will ask you to break the changes apart into multiple pull requests. They'll have an easier time reviewing the changes in small batches.

## Schedule the merge and publish of a release branch

1. At least one week before your release, [file a ticket for the content production team that specifies](#):

- That you need a release branch merged to master in the azure-docs-pr repo.
- Specify the release branch you want merged.
- List the date of publication, the time you want the release branch published (10:00 AM or 3:00 PM, Monday - Friday).
- If you need an out-of-band (OOB) publish, list the time/date and provide a business justification.

The one week timeline assumes you're publishing on a standard business day during standard business hours (8:00 AM to 5:00 PM, Pacific). We don't publish on Saturdays and Sundays. You must surface unique release requirements more than a week ahead. As soon as you're aware of a unique requirement, file a ticket immediately. In the ticket, demonstrate the business justification and resources will have to be specially scheduled to support a release outside standard days and hours.

2. The content production team acknowledges the work item.

The team requests approval from the business approvers if there's an OOB publish request.

3. By noon the day before the release, the content production team creates a pull request from the release branch to the master branch to get a build test. If merge conflicts are present and minor, the team fixes the merge conflicts. If the team can't resolve the merge conflicts because of complexity or because the technical meaning of content is at risk, the team will notify the author. That person is then responsible for fixing the issues in the release branch.

4. All content must be merged to the release branch by noon Redmond time on the business day before the release.

5. After noon the day before the release, the content production team creates a pull request from the release branch to the master branch to get a build test. If merge conflicts are present and minor, the team fixes the merge conflicts. If the team can't resolve the merge conflicts because of complexity or because the technical meaning of content is at risk, the team will notify the author. That person is responsible for resolving the issues in the release branch.

6. Before 5:00 PM on the business day before the release, the author must update the ticket with a comment to:

- Confirm the release date and time.
- Update it if they delay the release.
- Close the request if they cancel the release.

If a comment confirming the release isn't received, the team will hold off publishing to avoid any potential disclosure of embargoed content.

7. The content production team merges the pull request from the release branch to master per the date/time in the work item.

8. After the branch is merged, the content production team deletes the release branch in azure-docs-pr. Don't reuse branches in azure-docs-pr once they've been published.

If the content production team doesn't delete a release branch, file a ticket using the template in step 1 of this section to request its manual removal.

9. If pending pull requests still exist against the merged release branch, the team will close the pending PR. They'll add a comment in the closed PR telling the author they merged the branch. Then the team will delete the merged release branch.

## Release planning process for large releases

For large events like Build, Ignite, and Connect, the involved parties choose someone to act as the release manager. The team creates detailed release plans and the release manager shares the plans across the Content & Learning organization. For these releases:

- Request a release branch, [add release branch to the release tracker](#).
- Individual work items for publishing aren't required - the release plan will account for merging all the branches for the release.

## Best practices - releases involving multiple contributors and a release branch

- Releases that are timed/scheduled and that involve multiple authors should be handled with a release branch and a release plan. PRs will be merged along the way into the release branch.
- Define a content complete date. Content is complete when:
  - Author signs off on PRs.
  - PR review team reviews PRs.
  - Author addresses review feedback.
  - PR review team merges all PRs.
- Then, work back to define a date by which all PRs must be submitted and signed off. Allow 2-3 days before the content complete date.
- Large releases (BUILD, IGNITE) may require a content freeze of 2-4 days to stabilize the content set and release process. To ensure that content that you submit late doesn't pose problems, a process for triaging late PRs can help. You can tag PRs that aren't essential with labels for merge after the main event publishing run occurs.

## Iterating quickly in release branches/requesting exceptions to the PR review process

- If an author wants to stub out new articles on a one-time basis, the author should contact the repo admin to request the PR for the stub articles be merged, bypassing the PR review process.
- To support iteration over a period of a few weeks, the repo admin can choose to give trusted authors write perms in a repo. The master and live branch in the repo must be protected so only the PR reviewers, the admin, and Pubdesk (if applicable) can merge to master. For live, only the people who run publishing in your repo should have access, the PR reviewers shouldn't. Authors should contact their repo admin.  
After the repo admin grants write perms:
  - The authors are then allowed to merge freely into their release branch up to two weeks before the release, at which point the admin removes their write perms.
  - At this point, the PR review team reviews a PR from the release branch to master so they can see the diffs. They provide feedback to bring the release branch into compliance with the PR review criteria.
  - The author(s) have 2-3 days to address the feedback.

- The workflow for PRs goes back to the normal workflow, and the PR review team reviews all PRs that require human review.

## Release plan template

- Coordinate with your PR reviewer early on to validate that your release plan is feasible.
- For `azure-docs-pr`, coordinate with [techdocprs](#) and [pubdesktech](#).
- Confirm details with them at least a week before the release date. These gates are the tightest permitted release gates.

DAY	TIME	ACTION OR DEADLINE
2-3 business days before the release		All PRs signed off and ready for review by PR review team.
Business day before release	noon	Content complete in the release branch at noon.
		Content complete = PR signed off, reviewed by PR review team, feedback addressed, and PR merged.
		Submit your final PRs 2-3 days ahead of content complete.
		Author creates the PR to master and lets it build to ensure there are no conflicts, errors, or warnings. Fix any issues.
Day of release	8:00 AM - 9:00 AM	PR reviewer merges the release branch to master, and then deletes the release branch.
Day of release	10:00 AM	Typically, releases align to the standard publishing run at 10:00 AM Pacific Time.
		If you've tied this release to another event, you can request an OOB time. Qualifying events include an announcement event, a conference, or a planned blog post.
		The OOB time should be two hours before the time you want the content available. Remember to account for time zone differences.

## Complex release plans

Events like Build and Ignite involve *hundreds* of contributors and *thousands* of content updates and additions. Special events require release plans that are more detailed and that span many weeks leading up to the event date. Here are links to two recent release plans for these events:

- [Build 2017](#)

- Ignite 2017

# New contributor checklist for new content

4/16/2021 • 4 minutes to read

This article provides content contributors a checklist of tasks on how to plan, write, and publish a new content set on docs.microsoft.com, for a new product or feature.

## NOTE

This article assumes you have completed the contributor training. If you haven't please complete the [contributor training form](#).

You may want to review the overall guidance about onboarding content to [docs.microsoft.com](#).

## Plan

<input checked="" type="checkbox"/>	TASKS
<input type="checkbox"/>	Identify the lead PM, sponsor, and subject matter experts for the new content.
<input type="checkbox"/>	Identify the expected timeline for the release.
<input type="checkbox"/>	Identify if a repository for your need exists. If not, <a href="#">request the creation of a content repo</a> .
<input type="checkbox"/>	Work with your product team to understand customer challenges or user scenarios to help determine the content that you want to create. Review the <a href="#">writing principles</a> and the <a href="#">acceptable content for Microsoft Docs</a> .
<input type="checkbox"/>	Consider whether you'll need <a href="#">reference content</a> for any APIs, libraries, packages, or CLIs. Onboarding times can be long, so take that into consideration.
<input type="checkbox"/>	Determine if the release is public preview or general availability. For a preview, see <a href="#">Identify preview content</a> , and review how it will affect your table of contents and landing page.

## Design

<input checked="" type="checkbox"/>	TASKS
-------------------------------------	-------

<input checked="" type="checkbox"/>	TASKS
<input type="checkbox"/>	<p><i>If a repo doesn't exist:</i> Work with your product team and marketing to determine the service slug value. The service slug is the part of the docs.microsoft.com URL that identifies your service. The service slug is used for the repo folder or subfolder name. It's the value of the ms.service metadata attribute in articles.</p> <p>For example, <i>virtual-machines</i> is the service slug for Azure Virtual Machines and is consistently used between the Docs site (<a href="https://docs.microsoft.com/azure/virtual-machines">https://docs.microsoft.com/azure/virtual-machines</a>) and the ACOM site (<a href="https://azure.microsoft.com/services/virtual-machines/">https://azure.microsoft.com/services/virtual-machines/</a>).</p>
<input type="checkbox"/>	<p><i>If a repo already exists:</i> Work with the sponsor to coordinate your new content.</p>
<input type="checkbox"/>	Create a <a href="#">new release branch</a> for your new content.
<input type="checkbox"/>	<p>By using the new service slug value, create a corresponding folder to house your new content. Create the folder locally, and add your first files. The folders are "created" in GitHub when you add the first file and push it to the online repo.</p> <p>For example, this folder matching the service slug is used for the Azure Virtual Machines articles:  <a href="https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines">https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines</a>.</p>
<input type="checkbox"/>	<p>Create the <a href="#">new metadata value</a>. Use the <a href="#">new ms.service value request form</a> to add the new metadata to the list. Be sure to select <b>Yes</b> to include your new Docs metrics in the <a href="#">content engagement reports</a>. The ms.service metadata property typically uses the same value as the service slug.</p>
<input type="checkbox"/>	Get the new .svg icon from the marketing representative for your release or product.
<input type="checkbox"/>	Based on your content needs, list the articles that you need to create. Review the recommended <a href="#">article types</a> for Docs.
<input type="checkbox"/>	Based on your article list, <a href="#">define the TOC structure</a> to develop the toc.yml file. If a related toc.yml file is available, copy it into your new content folder and change it based on your article list. Use the <a href="#">TOC checklist spreadsheet</a> to draft your TOC.
<input type="checkbox"/>	Start thinking about <a href="#">landing pages</a> and <a href="#">hub pages</a> for your content. If a similar index.yml (landing page) file is available, copy it into your content folder and change it to suit your needs.
<input type="checkbox"/>	Add your content set to the main breadcrumb file: \bread\toc.yml. Review the <a href="#">breadcrumb guidelines</a> .

	TASKS
<input type="checkbox"/>	If you have a product forum URL, add it to the docset's docfx.json file: MicrosoftDocs/docs-help-pr/docfx.json. Under <code>feedback_product_url</code> , add your folder path and feedback URL. (See other line items as examples.) Review the article <a href="#">How to enable the documentation feedback control</a> .
<input type="checkbox"/>	<a href="#">Submit a request</a> to add the new content on your area hub page (if it applies), and include the new icon to use.

## Write

	TASKS
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	Ensure your articles align with the docs.microsoft.com <a href="#">writing</a> and <a href="#">technical principles checklist</a> .
<input type="checkbox"/>	Understand the various <a href="#">content types</a> before you create your content.
<input type="checkbox"/>	Understand and incorporate <a href="#">SEO techniques</a> in your content.
<input type="checkbox"/>	Ensure your content has all the required <a href="#">Metadata</a> .
<input type="checkbox"/>	<a href="#">Run Acrolinx</a> to ensure compliance with Microsoft standards.
<input type="checkbox"/>	Create your <a href="#">TOC</a> .
<input type="checkbox"/>	Create your <a href="#">Landing page</a> .
<input type="checkbox"/>	Divy Sharma (DIVYARATNA) is responsible to include your new content. Confirm with Divy Sharma that everything is on schedule.

## Review and publish

	TASKS
<input checked="" type="checkbox"/>	
<input type="checkbox"/>	When your content is ready, understand the <a href="#">pull-request (PR) submission recommendations</a> and create a PR to receive feedback from SMEs.
<input type="checkbox"/>	Check the build validation, and address any issues.
<input type="checkbox"/>	Check the Acrolinx results, if available. See <a href="#">Acrolinx coverage</a> .
<input type="checkbox"/>	<a href="#">Invite reviewers</a> to provide feedback on the new content.

	TASKS
<input type="checkbox"/>	After you incorporate the reviews and feedback, sign off ( <b>#sign-off</b> ) on your PR to merge it. If you're using a release branch, you might create several PRs to bring in content from all contributors. Review each PR before merging with the release branch. Signing off is a requirement for Microsoft Docs. Talk to your repo admin for your process.
<input type="checkbox"/>	Schedule your <a href="#">release branch merge</a> with the main branch.
<input type="checkbox"/>	Request <a href="#">Go Live</a> .
<input type="checkbox"/>	Generate a <a href="#">forward link (FWLink)</a> to your landing page so your product team can link to the content from their product's user interface.
<input type="checkbox"/>	Check if your content requires a Product Launch Readiness (PLR) sign-off. PLR is a process where the product and the articles require a sign-off for the public preview or GA stage. For Azure, remember to request a <a href="#">PLR sign-off</a> for articles.
<input type="checkbox"/>	After the content is published, do a visual pass of all the content, including any hub and landing pages.

## Maintain

	TASKS
<input checked="" type="checkbox"/>	Validate accuracy. The longer content has been published on the live site, the more likely it needs a review for accuracy. Dependencies, brand names, and metadata can change over time.
<input type="checkbox"/>	<a href="#">Review content performance</a> .
<input type="checkbox"/>	Address any <a href="#">GitHub issues</a> .
<input type="checkbox"/>	<a href="#">Retire old content</a> .

# How content resources are assigned to product documentation

4/16/2021 • 9 minutes to read

Delivering technical documentation depends on strong partnerships between content development and product teams. Each team has a part to play. This article describes the content support that the Developer Relations Content & Learning team provides and also the responsibilities and expectations of product team partners. Specifically, these guidelines apply to product documentation maintained by Martin Ekuan's organization.

The Content & Learning team provides content development resources to some products, services, and technologies, and centralized support for all products, services, and technologies.

## Resourcing criteria

When a product is funded and staffed at the product group level, there is often no allocation of resources for content development. Until we get content development resources to be part of the initial finance and staffing discussion, the Content & Learning team will work with teams to staff projects ad hoc.

The Developer Relations Content & Learning team divides products and services into three groups and assigns resources accordingly. Content & Learning uses the following criteria for assignment:

- **Cloud first:** Cloud services and workloads that prioritize moving customers to the cloud are typically prioritized over on-premises only products.
- **Microsoft strategic direction and big bets:** Product or service is crucial to delivering on Microsoft big bets and strategic direction.
- **Clear development roadmap:** New services and/or features are planned, funded, and scheduled.
- **Product team availability:** Product teams can contribute to planning, creation, and maintenance. The product team engineers and PMs are committed to content planning, creation, and maintenance.
- **Customer/install base:** Current and projected.
- **Revenue:** Current and projected.
- **Complexity:** More complex services/products require more attention to deliver a positive customer experience.
- **Azure Ring assignment:** Ring 0 is often prioritized above Rings 1 or 2, but Azure Ring model does not directly correlate with our resource assignment.

## Support model

The following table summarizes the different services provided by Content & Learning and three basic models to group services. While roles and responsibilities vary from each relationship, the three models provide a baseline for discussions. For "self-managed" relationships, the product team authors and maintains the content themselves; for the other models, C&L authors and maintains content to various degrees, in shared content creation relationships.

	BROAD	PARTIAL	SELF-MANAGED
--	-------	---------	--------------

	BROAD	PARTIAL	SELF-MANAGED
<b>Criteria</b>	<ul style="list-style-type: none"> <li>- Cloud first</li> <li>- Strategic or big bet</li> <li>- Clear development roadmap</li> <li>- Engineering team availability</li> <li>- High customer/install-base/revenue</li> <li>- Available content team resources</li> </ul>	<ul style="list-style-type: none"> <li>- Strategic or big bet</li> <li>- Clear development roadmap</li> <li>- Engineering team availability</li> <li>- Medium customer/install-base/revenue</li> <li>- Available content team resources</li> </ul>	<ul style="list-style-type: none"> <li>- Development roadmap unclear</li> <li>- Or low/no engineering team availability</li> <li>- Or low customer/install-base/revenue</li> <li>- Or lack of content team resources (<a href="#">Can you fund?</a>)</li> </ul>
<b>Content Strategy</b>	✓	✓	✓
<b>Shared Content Creation</b>	✓	✓	✗
<b>Onboarding + Training</b>	✓	✓	✓
<b>IA/TOC/Hub + Landing pages</b>	✓	✓	✓
<b>Editorial Service</b>	✓	✓	✓
<b>Content Dashboards</b>	✓	✓	✓
<b>Quarterly Reports</b>	✓	✓	✓
<b>Monthly Sprint Reports</b>	✓	✓	✗
<b>Top Scenarios + Tasks</b>	✓	✓	✗
<b>Release Management</b>	✓	✓	✗
<b>Customer Research</b>	✓	✓	✗
<b>Active Monitoring and Improvements</b>	✓	✓	✗

The following sections describe the responsibilities of the content developer and the product team for each support model:

## Broad support

The Content & Learning team assigns one or more content developers to a service to work with the product team to develop content.

### Content developers' primary responsibilities:

- Develop and manage the service's overall content strategy and content work prioritization on a sprint-based cadence.
- Provide onboarding assistance and content release planning for new products and services
- Work with product team partners to identify and prioritize [top customer scenarios or tasks](#) for the service.
- Information architecture work, including TOC and landing/hub page design.

- Create and maintain accuracy and freshness of owned content.
- Respond to and resolve Git issues and public pull requests for owned articles.
- Monitor documentation health and performance; recommend and prioritize corrective actions.
- Gather and synthesize qualitative and quantitative customer feedback; recommend and prioritize corrective actions.
- Work with vendor editing resources to schedule and perform documentation edits.
- Orchestrate and manage documentation release events for service and feature releases, event-driven releases (Ignite, Build, etc.)
- Attend marketing onboarding, greenlight sessions, etc.

#### **Product team's primary responsibilities:**

- Assign one person as primary content contact who meets regularly with the content team lead.
- Identify, validate, and help prioritize service's top customer scenarios, share insights from user research, experimentation, and other customer connections.
- Create (to content model specifications), update, and maintain the accuracy and freshness of owned articles.
- Respond to and resolve Git issues and public pull requests for owned articles.
- Provide source information and technical reviews for articles that the Content team writes.
- Use Content & Learning's [Azure DevOps project](#) and sprint cadence to track content work.
- Include content developers and/or managers in service and feature planning.
- Include content developers and/or managers early on in release planning.
- Include content development in blog and social media planning.
- Include content developers and/or managers in green-light planning.
- Add content developers and/or managers to relevant communication channels.
- Develop and maintain all reference documentation.

## **Partial support**

Content & Learning assigns one content developer, part time, to a service to work with the product team to develop content.

#### **Content developers' primary responsibilities:**

- Develop and manage the service's overall content strategy and content work prioritization on a sprint-based cadence.
- Provide onboarding assistance and content release planning for new products and services.
- Work with product team partners to identify and prioritize [top customer scenarios or tasks](#).
- Information architecture work, including TOC and landing/hub page design.
- As time permits, the content team might share ownership of overview, quickstarts, tutorial, and documentation targeted to the top scenarios for that service.
- Respond to and resolve Git issues and public pull requests for owned articles
- Monitor documentation health and performance; recommend and prioritize corrective actions.
- Gather and synthesize qualitative and quantitative customer feedback; recommend and prioritize corrective actions.
- Work with vendor editing resources to schedule and perform documentation edits.
- Orchestrate and manage documentation release events for service and feature releases, event-driven releases (Ignite, Build, etc.).
- Attend as time permits, appropriate planning meetings, marketing, and greenlight sessions.

#### **Product team's primary responsibilities:**

- Assign one person as primary content contact who meets regularly with the content team lead.
- Identify, validate, and prioritize service's top customer scenarios.
- Create (to content model specifications) and maintain accuracy and freshness of owned articles (which might include overview, quickstarts, tutorials, examples, conceptual, and how-to documentation).
- Respond to and resolve Git issues and public pull requests for owned articles.
- Provide source information and technical reviews for articles that the Content team writes.
- Use Content & Learning's [Azure DevOps Boards instance](#) and sprint cadence to track content work.
- Include content developers and/or managers in service and feature planning.
- Include content developers and/or managers early on in release planning.
- Include content development in blog and social media planning.
- Develop and maintain all reference documentation.

## Self-managed support

You have no dedicated content developer resources. Instead, you have a single point-of-contact on a relevant content team who can answer questions and provide some guidance. You also have an assigned Content Manager who can help with complex issues and escalations that are not addressed by other resources. Your Content Manager will also approve changes to landing/hub pages. Self-managed support includes the following resources:

### Self-managed support resources:

- [Docs contributor program](#) for onboarding and support for teams without a dedicated content development resource.
- [Contributor Guide](#): Comprehensive guide to content development.
- Tools for writing and publishing: [VS Code extensions](#), [API reference generation tooling](#), Open Publishing System (OPS), build validation, and staging environments.
- Training: [Using GitHub](#), authoring tools and templates, publishing process, applying the [content model](#), style guidance; how to "do" content. The centralized team for the docs.microsoft.com (Docs) Contributor Program provides core training sessions and materials.
- [Editorial support](#): Centralized vendor editorial service that can copy edit articles after the content is published live to docs.microsoft.com.
- Ad hoc support: Teams channel, presentations, and [office hours](#). More complex support issues are escalated to [Content Production Service \(CPS\)](#).
- [Localization services](#).
- Terminology and branding guidance and consulting.

### Product team's primary responsibilities:

- Identify, validate, and prioritize service's top customer scenarios.
- Information architecture work, including TOC and landing/hub page design in alignment with documented guidance.
- Own all service/product documentation, which includes aligning with content model specifications and maintaining accuracy and freshness of all owned articles.
- Work with vendor editing resources to schedule and perform documentation edits.
- Develop and maintain all reference documentation.
- Fund vendor writers or specialized editors, as needed. Content Manager can assist with hiring recommendations.
- For those groups with self-managed docs.microsoft.com repositories, transfer repository management to the vendor-supported repository administration service.
- Respond to and resolve Git issues and public pull requests for owned articles.

- Monitor documentation health and performance, apply corrective actions per recommendations from centralized team.
- Gather and synthesize qualitative and quantitative customer feedback; recommend and prioritize corrective actions.
- Optional: Use Content & Learning's Azure DevOps project and sprint cadence to track content work.
- Manage content planning for upcoming features, releases, social media, and blog posts.
- Include your Content Manager in major planning and greenlight reviews.
- Partner with your Content Manager when preparing quarterly business reviews with product leadership.

## What is article ownership?

The article owner is whoever is responsible for a given article and is responsible for keeping the content accurate and up-to-date. The owner is typically a content developer or a product team contributor and is the listed author in the article's metadata. Article owner expectations.

- Create and update articles, following [content model and style guidelines](#).
- Request editorial reviews and approve article editing work with the [vendor editorial pool](#).
- Respond to and resolve Git issues filed against their articles within the [defined SLA](#).
- [Review, comment on, accept or reject public pull requests \(PR\) submitted for their articles](#).
- Keeps articles [current and accurate](#).
- Resolve bugs filed against articles in Content & Learning's [Azure DevOps project](#).
- Monitor article [health and performance](#).

## Options for additional resource funding

In the past, some teams have hired vendor or FTE content developer resources that remain organizationally aligned with the product team. This organizational alignment under management of the product team has led to publishing and content quality issues. As such, The Content & Learning team has established a policy that **content development resources (FTE or vendor) must be managed by a Content Manager in Content & Learning**.

If you work on a project where more content development resources are needed, there are several options:

- Transfer PCN or funding to the Content & Learning team. When PCN or funding transfer occurs, that headcount is ring-fenced for the project. Product teams are guaranteed that the resource stays with the project.
- Members of the team (Program Managers, engineers, among others) write the technical content in addition to their typical job responsibilities. The team remains in centralized support.

## New products or services

The Content & Learning team reviews resource assignments quarterly. Contact your assigned Content Manager or the documentation team as soon as you are aware of a service or product release that will require documentation.

## Next steps

If you require content support for a new product or service, reach out to your existing contacts in Content & Learning:

- [List of business approvers and repositories](#)
- [Business approver list for Azure services](#)

If you don't have any contacts, use the following resources:

- [Docs Contributor Program](#)
- [Docs.microsoft.com content onboarding request](#)

# Use release branches

6/1/2021 • 13 minutes to read

This article describes using release branches in Git and GitHub for planned releases.

To use release branches effectively, do these tasks:

- Understand release branches
- Check out the branch
- Push branches
- Use multiple working branches
- Clean up branches
- Compare branches

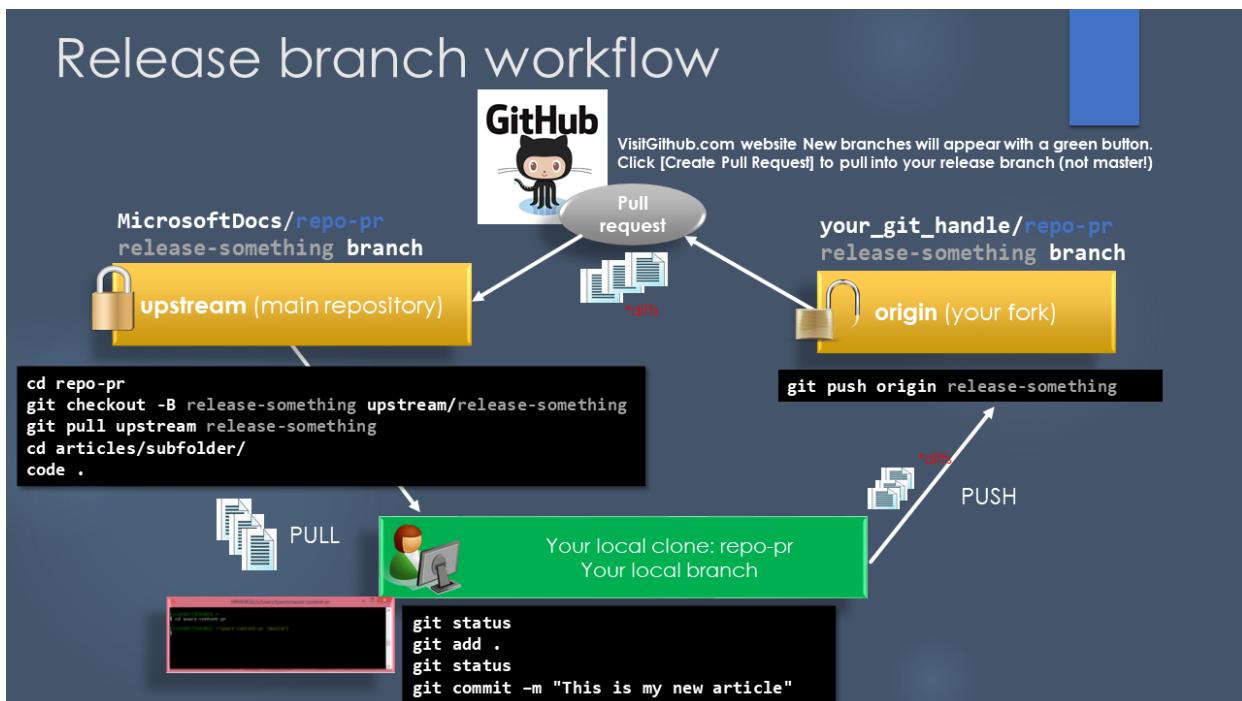
## Understand release branches

A release branch is a Git branch that a team or author creates to house documentation that they need to hide from public view. As such, a release branch is isolated from the master branch. The reason for creating a release branch varies. The release could be targeting a new product version or a certain release date. The release date might be associated with a conference, a public preview announcement, or a general availability announcement. Typically, PubDesk publishes the master branch of a Git repo to the web on a [regularly scheduled cadence](#). A release branch, however, is kept internal and isn't published until a specified date.

**Do you need a new release branch?** To create a new release branch, make a request following the process outlined in [Content release planning + process](#). Plan ahead with PubDesk and the PR review team to ensure your content meets the quality bar and is published at the appropriate time.

Usually, PubDesk names the release branch like this `release--` or simply `release-`.

This graphic illustrates the pattern used when using working with release branches:



## Prerequisites

This article assumes you have already installed Git locally and cloned the repo by following these articles:

- [Install content authoring tools](#)
- [Set up Git repository locally for documentation](#)

It also assumes the release branch already exists. If not, you must [request a new release branch](#).

## Check out a release branch

When you're working with a [release branch](#) in the main upstream repo, you need to create a local copy of that release branch. Using your local clone repo, follow these steps:

1. Make your local clone aware of the new upstream release branch. Use Git Bash locally, and run the fetch command:

```
git checkout master  
git fetch upstream
```

The output from the fetch command lists new branches with a label [**new branch**].

The current active branch is teal and listed in parenthesis in the Git Bash prompt. Git Bash should list the default branch as `(master)`. If there's no active branch shown, you need to change directory into the correct repo subfolder.

2. View the list of branches by running the `git branch` command:

With no parameter, the branch command shows branches on your local machine in the current repo.

```
git branch
```

The Git Bash output lists the current active local branch in green and it's flagged with an asterisk.

Repeat the command using the `-r` parameter, to see the remote branches in the GitHub site (both origin and upstream locations) that are available to check out.

```
git branch -r
```

Locate the remote upstream/\* branch for your release branch.

3. Create a local branch that tracks the appropriate upstream release branch. This command creates the local branch directly by tracking the upstream branch, and resets the local branch to match without merging.

For ease and convenience, this approach uses the same release branch name in both places: your upstream fork and your local Git repo.

```
git checkout -B <release branch name> upstream/<release branch name>
```

Example:

```
git checkout -B release-build-some-service upstream/release-build-some-service
```

For advanced scenarios, you can customize the local branch name. You may decide to give your local branches purposeful and friendly names. It's easier to remember what task you're working on when you

name your branch `new-quickstart`, `editing-tutorial`, and so on. Later on, when you push, you need to be more aware of the branch name differences because it doesn't match by default.

```
git checkout -B write-new-quickstart-locally upstream/release-build-some-service
```

4. At least daily, keep your local copy of the release branch (and any working branches derived from it) up to date with the upstream branch.

```
git pull upstream <release-branch-name>
```

#### WARNING

Notice that you don't want to pull from `master` into your release branch.

The main release branches are typically refreshed daily. PubDesk carries out a one-way merge in from the master branch. This merge is done to keep release branches current, and contain the contributions made by the team at large. However, the release branch isn't merged back into master. The changes in the release branch are kept private in isolation until the actual release time frame.

5. Now you're ready to write your content.

## Commit and push a release branch

Once you're done working on an article locally in a release branch, inspect and commit the changes locally as you would normally. Then push to the release working branch in your fork.

For long running projects, do the push into GitHub periodically, so that you have a safe backup of your local branch in your cloud side fork.

1. Inspect changes and add the updated files:

```
# check the status output to make sure the files listed are as you expect. Red shows new or changed files.  
git status  
  
# add the files listed. wildcard adds all under the current path.  
git add .  
  
# check the output to make sure the expected files are now green  
git status
```

2. Commit the changes locally:

```
git commit -m "Comment for a set of changes. New article, etc..."
```

3. Push your local copy of the release branch to your online fork.

This example command assumes the branch names are identical:

```
git push origin <release-branch-name>
```

If there's no existing release branch in your fork, the system generates a new branch using the given `<release-branch-name>`.

If you've named your local branch with a custom name, you can specify both the local branch name and the remote fork branch name. You can vary the name on your fork when you push as needed. It's most convenient to use the release branch name. When you do so, pull requests (PRs) are faster and easier to initiate in GitHub because the name matches the release branch.

#### IMPORTANT

If your local branch name matches that of the upstream release branch, the branch name will be auto populated in the GitHub interface. If you created your local branch with a custom name, you will need to select the base branch in the GitHub interface.

```
git push origin <local-branch>:<new-fork-branch>
```

For example, use the release branch name on your fork for convenience:

```
git push origin new-quickstart-article:release-build-service
```

Use a purposeful name on your fork for multitasking:

```
git push origin new-quickstart-article:new-quickstart-article
```

## Propose your changes to the upstream repo

#### TIP

For the most updated instructions, go to the [Creating a pull request](#) topic in the GitHub documentation.

- Once the changes are ready, issue a PR.

Using the web browser, go to your fork on GitHub and create a PR to merge from your own fork into the main release-branch.

For example, the URL resembles:

```
https://github.com/<your-github-alias>/<repo-name-pr>/tree/<release-branch>
```

GitHub represents recent new branches created in your fork with a yellow alert bar and green button to find them with ease.

Your recently pushed branches:

 JasonWHowell:release-ignite-aml-v2 (less than a minute ago)



#### IMPORTANT

If your local branch name matches that of the upstream release branch, the branch name will be auto populated in the GitHub interface. If you created your local branch with a custom name, you will need to select the base branch in the GitHub interface.

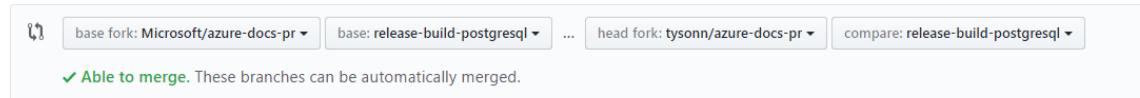
When you're ready to merge your changes, you create a PR from the release branch in your fork to the upstream release branch. The PR should be set up as shown in the screenshot.

## WARNING

Notice that you do not want to merge with **master**. If you do, PubDesk may publish the release branch changes prematurely.

## Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also [compare across forks](#).



## Using multiple working branches

If you're editing a handful of articles (less than 10), one working branch may be enough. However, if you're creating vast amounts of new content for a given release, isolate your work into batches using multiple working branches. Make a new working branch to keep changes discrete and isolated.

Use multiple working branches to batch changes. Avoid large PRs with more than 10 markdown articles. They're too large for the feedback process and become too complex to merge. When using multiple working branches, you can push each branch to your fork independently as batches. Then, you can have them merged into the main release branch by using an independent PR.

Use this workflow for creating multiple working branches:

1. Inspect any uncommitted changes.

```
git status
```

Notice any in-flight uncommitted changes before you switch branches. The changes listed from `git status` (in red) follow you as you switch branches. As such, it's easy to accidentally put a change in the wrong branch. Add and commit such pending changes before checking out a different branch to avoid that confusion.

2. Create multiple working branches from your local copy of the release branch.

This series of commands gives you two working branches derived from your main local release branch:

```
# checkout your local release branch
git checkout <release branch>
git pull upstream <release branch>

# make a working branch based on the local release branch
git checkout -b <working-branch-1>

# make another working branch based on the local release branch
git checkout <release branch>
git checkout -b <working-branch-2>
```

## NOTE

If your source local release branch has recent changes, the two other branches would duplicate those changes as well. You may use `git checkout -B <new-local-branch-name> upstream/<release-branch-name>` to get secondary copies of the release branch (minus any pending PRs) from the main branch in the GitHub site.

3. As needed, you can switch between multiple branches using the checkout command:

```
git checkout <working-branch-1>
## do a batch of work

git checkout <working-branch-2>
## do additional isolated work
```

When viewing the files in the directory on your local disk, Git represents the files available in the current branch only. As you switch branches, the list of files changes to match. This behavior applies to the files seen in Windows Explorer or when using an `ls` or `dir` command. The branches are isolated from one another logically and Git enforces that on the file system. When you switch branches to

`<working-branch-2>`, Git removes local files unique to `<working-branch-1>` from the local disk and any files unique to `<working-branch-2>` are visible. When you checkout `<working-branch-1>` again, those files reappear. That's how Git isolates the changes so you can work locally.

4. At least daily, keep your local copy of the release branch (and any working branches derived from it) up to date with the upstream branch.

```
git pull upstream <release-branch-name>
```

PubDesk runs a process that automatically updates release branches with updates from the master branch on a daily cadence.

**TIP**

If you receive a **fatal: Cannot update paths and switch to branch 'release-branch' at the same time** error when issuing the `checkout` command, run `git fetch upstream`, then rerun the `checkout` command. `fetch` grabs all the new remote-tracking branches (such as the release branch you want to work with) and tags without merging those changes into your own branches.

## Cleaning up branches periodically

To avoid confusion when maintaining multiple branches, it's best to clean up your local branches and the branches in your remote fork. Keeping a tidy repo helps keep you focused. It also removes unnecessary clutter that becomes confusing when working on multiple tasks.

Once the PRs for a given branch are all merged or closed, you can clean up the local branch. Start with a fresh new local branch for the next batch of work you need to do. You can delete the local branches that you no longer need.

1. Check out the master branch when you need to remove a local branch:

```
git checkout master

#list local branches
git branch
git branch -d <mylocalbranch>
```

Use `-D` cautiously when prompted by the system to remove uncommitted or unmerged changes.

2. Inspect and delete remote branches in your fork.

- a. For branches in your fork, check for any outstanding PRs before deletion. GitHub doesn't issue any warnings when removing branches and automatically closes any outstanding PRs originating from

a deleted branch.

GitHub search syntax:

- *is:open is:pr head:<branch-name> repo:MicrosoftDocs/<repo>*
- *is:open is:pr base:<branch-name> repo:MicrosoftDocs/<repo>*

b. You can use the web browser or Git Bash command line to delete branches in your fork.

In the browser, go to the URL for your fork and see the branches for a specific repo:

```
https://github.com/<your-github-alias>/<repo-name>/branches
```

To delete a branch, select the trash can icon beside a branch.

Notice that when you first forked the repo, GitHub lists all the branches that were available on your local fork. You may remove those extraneous branches (except the default master branch) from your fork to tidy up.

Branches that you made with the GitHub web editor have the naming scheme **patch-##**.

c. Another option is to remove branches in the Git Bash prompt using `-d` to delete and `-r` for remote.

```
# list remote branches - origin/* are in your fork  
git branch -r  
  
# delete a remote branch from the origin location  
git branch -dr origin/<branch-to-delete-from-fork>
```

#### WARNING

Do not attempt to remove upstream branches, as PubDesk manages them. Most contributors have read-only access to the upstream repo and cannot directly make changes there.

## Be cautious when switching between branches and creating branches

The point of release branches is to allow you to isolate content for release on a specific future day. Lack of care in creating branches locally and in switching branches can result in content confusion!

- If you are in the master branch, and you have added or committed new changes and content, and you then use one of the following commands to create a new local branch, the new branch contains your added and committed changes because these commands create the new branch from the current branch:

```
git checkout -B <new-branch-name>  
git pull upstream master:<local-working-branch>
```

- Before you switch between branches, always run `git status` to see if you have unadded or uncommitted changes. If you have outstanding changes in a branch, and you switch branches, your uncommitted changes float with you and are committed in whichever branch you happen to run the `commit` command next.

## Comparing a release branch to master

For quality assurance purposes, you may inspect the list of new files in the release branch to make sure it only

includes the changes you expect.

When you know that your release branch is up to date with master, you can create a pull request from the release branch to the master branch. The list of files that appear in the diff should be the new content for the release. Always QA the list of files before you merge to master to ensure your release branch modifies *only* those files you intended to change. It's common for release branches to end up with extraneous, unintended changes. This happens most often when significant content refactoring has occurred.

It's common for release branches to contain more than 100 changed files. In this case, you can't use a pull request to view all the diffs to ensure the changes are intentional.

You can use this process to check the list of files and diffs:

1. Make sure your upstream release branch is up to date with master. File a pull request from master to your release branch to accomplish this update.
2. In your local clone, in master, pull the upstream content:

```
git pull upstream master
```

3. Check out your local copy of the release branch, and pull the upstream release branch content:

```
git pull upstream <release-branch>
```

4. Run this command to generate a list of all the files in your branch that are different when compared to master:

```
git diff --name-only master <release-branch>
```

5. As a best practice, run the same command the other way to see what master might have that your release branch doesn't:

```
git diff --name-only <release-branch> master
```

6. If there are files in your release branch that you didn't intend to change, restore those files by bringing them back over from master.

Run the `checkout` command against individual files or against entire folders in the repo:

```
git checkout master <path-to-file/or-folder/ending-with-articles.md>
```

7. Run `git status` to confirm that the `checkout` command is working, then add, commit, and push your changes when done.
8. Create a pull request against your release branch to bring these changes in.
9. Repeat the QA steps until the list of diffs contains only the files you intended to change.

## Next steps

[Writing content for Content & Learning](#)

# Identify preview content

5/21/2021 • 2 minutes to read

*Preview* refers to pre-release products, services, and features that are made available to customers for evaluation purposes and at reduced or different service terms.

## Private preview

A service or feature in *private preview* is disclosed to only a select customer base. We do not include private preview content on docs.microsoft.com. Instead, the product team can provide accompanying technical documentation as a PDF file or privately hosted webpage.

## Public preview

A service or feature in *public preview* is available to the public, but not considered generally available (GA) or production ready and does not have a guaranteed SLA. A public preview might include preview or other pre-release features, services, software, or regions.

Public preview content is allowed on docs.microsoft.com. This includes content for *gated public previews*, where the product, service, or feature is publicly disclosed and available, but users must be added to an allowlist or request special provisioning before use.

The [Azure Preview feature webpage](#) lists some of the Azure services and features currently in public preview.

### Product and feature names

When referring to the name of a product, service, or feature in public preview, we follow the general guidance in the [Cloud Style Guide](#).

For product and service names, use *Product Name Preview* (preferred) or *Code Name Preview*. When referring to features, add *(preview)* after the name of the feature.

Examples:

- **Service:** Azure Data Explorer Preview
- **Feature:** Azure Ultra SSD (preview) delivers high throughput, high IOPS, and consistent low latency disk storage for Azure IaaS VMs.

### Headings, titles, and TOCs

Use the general naming guidance above to identify product, service, or feature previews in your headings. Also identify the preview by name in the article's `title metadata value`.

Do not add *Preview* or *(preview)* to your TOC labels.

### Body

Use the general naming guidance above to identify product, service, or feature previews on first mention in the body and in each major section. If context is clear, subsequent mentions in each major section can omit *Preview* or *(preview)*.

Avoid additional or generic use of *preview* in your articles, if possible. If you need to discuss functionality or limitations directly related to the preview release, use a separate article or section that is easy to identify and remove when the product goes GA.

If you want to call out the supplemental terms of use, you can add an alert at the top of each article or section

that discusses the preview. We recommend you use this standard language:

```
> [!IMPORTANT]
> <Product, service, or feature name without "preview"> is currently in PREVIEW.
> See the [Supplemental Terms of Use for Microsoft Azure Previews]
(https://azure.microsoft.com/support/legal/preview-supplemental-terms/) for legal terms that apply to Azure
features that are in beta, preview, or otherwise not yet released into general availability.
```

You only need to include the alert once per topic file. If multiple articles in your docset require the preview alert, you can create an [include file](#). In this case, be sure to include "preview" in your include file name.

## Removing preview language

When preparing a GA release of a service or feature in public preview, content owners are responsible for removing all mentions of the preview in the technical documentation. We recommend you search on "preview" across all source files in your docset, the `/includes` folder, and other areas of your repository to identify possible instances.

# Request approval for an exception

5/10/2021 • 2 minutes to read

Most of the content we create for Docs fits the standard Content & Learning content process for [articles](#), [TOCs](#), and [landing pages](#). Occasionally, content that doesn't adhere to the criteria needs to be published as an exception. **All exceptions require review and approval by Martin Ekuan.** Exceptions are either temporary or approved.

- **Temporary exceptions** are used when you don't have time to adhere to the published guidelines because of a deadline. Temporary exceptions require that the content that deviates from the current guidance is fixed in a short period of time.
- **Approved exceptions** are used when you have conducted [customer research](#) and have data to prove that the exception improves the customer experience for the article, TOC, or landing page. To be approved, exceptions must be reviewed Martin Ekuan and the C+L writing guidance must be changed to reflect the new outcome.

## Request an exception

For Azure content, you can file an exemption request for some standards in the [Content Standards report](#). Review the [list of exemptions available](#) in this report and use that process if it applies. Otherwise, to request an exception send email to Martin Ekuan ([Martin.Ekuan@microsoft.com](mailto:Martin.Ekuan@microsoft.com)). Explain whether you're requesting a temporary exception or an exception related to customer research.

If this is the first time content is being published for a product or service, you must create and submit a control article/TOC/landing page that adheres to the existing guidance as much as possible and an explanation of why this format isn't optimal for customers.

# Should I work in a private or a public repository?

5/21/2020 • 3 minutes to read

Most content on docs.microsoft.com is backed by files stored in GitHub repositories. Some of the docs repos have a public/private pairing that is synchronized regularly. You can choose which repo in the pair to contribute your docs changes to.

Content developers and core product team PMs should work in the private repositories so they can address quality feedback from the quality automation present in the private repos.

Microsoft employees from other teams can work in public repos at this time to ensure their contributions are reviewed by the article author. For example, if you work in CSS and only contribute occasionally, working in the public repository is lighter and easier for most docs contributions.

Microsoft employees outside the content team can work in the private repo as needed, and then request that the author review your changes using an @mention with their github handle.

## Why work in the private repo?

There are multiple reasons for content authors to work in the private repos:

1. A variety of quality support functionality is present in the private repos:
  - [Docs validation](#) runs on the updated article to check links, metadata values, and other basics.
  - Staging is enabled at the site review.docs.microsoft.com on most private repos. You should review all staged updates prior to publication to make sure the text formatting, headings, images, and table of contents work as expected.
  - You can use the `#sign-off` comment to indicate a pull request is ready for review and publishing.
  - Pull requests that contain minor, text-only updates to existing articles may qualify for automatic merging to master, significantly speeding up the time-to-publish.
2. If you are working on a time-sensitive release (such as a features leading up a conference, embargoed content, public preview launch, or general availability launch) use the private repository and a private [release branch](#) to protect that content until it is ready for release.
3. If you are updating an article where you are the listed author, use the private repository. The PRMerger system automatically closes pull requests in a public repo if you are listed as the author. Check the `author:` tag in the metadata at the top of the article.

## How to get to the private repo

One reason people end up in the public repo is because it's easy to click the "Edit on GitHub" link on the docs.microsoft.com article. To get to the private repo from there, add `-pr` to the repo name in the URL string, and you will see to the same article in the private repo.

The private repositories require you to sign in to GitHub with your account and have access to the repo. If you get a 404 error when browsing to the private repository that you know exists, try [signing in to GitHub](#) first. If you are already signed in, that 404 error typically means you do not have access. To get access, your GitHub account needs to be configured with two-factor authentication, and joined to the right the organization (such as [MicrosoftDocs](#)). Follow the steps in the [GitHub account setup](#) article.

### Example URLs

- Public repo URL ( `azure-docs` ):
  - <https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/firewall/overview.md>
- Corresponding private repo URL ( `azure-docs-pr` ):
  - <https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/firewall/overview.md>

## Move public changes by pulling to the private repo

If you made changes already in the public repo, you can move those changes to the private repo using Git Bash or another command-line tool for Git. Run the following command from your local master branch to pull the public repo content from your fork into the private repo:

```
git pull https://github.com/<your-GitHub-account>/azure-docs.git <the-branch-you-made-changes-in>
```

### Example

Bob made changes in the public repo in a branch called patch-124, Bob can pull those changes to his local clone's master branch. Bob's GitHub user name is GitHubBob, and the patch-124 branch is used:

```
git pull https://github.com/GitHubBob/azure-docs.git patch-124
```

Then, Bob does git add, git commit, git push origin, and creates a pull request to the private master branch.

# Editorial service for technical content

5/10/2021 • 4 minutes to read

To support basic quality in your content, Content & Learning provides an editorial service. There are currently two paths to an edit pass:

- You can request an editorial pass.
- The pull request review team may refer your content for edit if the content appears to need an edit pass when it goes through pull request review.

In both cases, the edit typically happens after publication.

## Overview of the editorial service

- The service is available to any technical article contributor, whether you're a content developer, program manager, or in another role.
- The service is available only for C + AI technical documentation.
- The service is available only for content that will be published to the documentation section of product websites or to TechNet, MSDN, or docs.microsoft.com.
- This service isn't intended for marketing content, blog posts, and other content that will be published outside a technical documentation library. If in doubt, contact msmbaldwin.
- This service is intended for individual article edits or small batches of edits. If you want to edit a collection of content (more than 10 articles), contact msmbaldwin and v-tamif to discuss.
- At this time, edits are focused on terminology, product names, spelling, punctuation, grammar, and basic Microsoft Style. The core style references are the [Microsoft Cloud Style Guide](#) and the [Microsoft Writing Style Guide](#).
- **Edits are made post-publication.** This best practice is based on experience. Content authors frequently make changes to content after pre-publication edits, so the content ships with copy edit issues anyway. Or, the parallel edit and authoring changes cause unresolvable merge conflicts and wasted effort by the edit team. Edits made a couple weeks after initial publication produce higher-quality results with less churn for both authors and editors.

## Exceptions for pre-publication edits

The editorial service can support limited pre-publication edits, under these conditions:

- The author cannot make changes during the edit period.
- The edit request is filed to allow adequate time for the edit and for a feedback loop with the author. Allow one working day per every 2,000 words for the edit and two working days for the feedback cycle between editor and author.
- There should be some business justification for doing the edit before publication – it is being featured in an executive presentation, Gartner will be reading it, or some other attribute that means the content is higher visibility than most content.
- The edit team must have available capacity.

## Some content may be referred automatically for copy edit

In the pull request workflow, pull request reviewers may file an edit request if any of the following problems are noted during the pull request review:

- Incorrectly spelled product names.
- Missing punctuation.
- Incorrect spelling, grammar, and punctuation or missing punctuation.

If any of these issues are present, the article will also be edited for alignment with basic Microsoft style.

## To request an edit pass

1. File a work request: <https://aka.ms/APEX-edit-request>.
2. In the work request, clearly explain the content you need edited – the ticket can be for one article or multiple articles. The ticket must include:
  - The name and URL to the GitHub repository.
  - If the repo is outside the Azure product area, include instructions on how the editors obtain read permissions to your private repository.
  - A list of the articles that you want edited - provide the full production URL of each article.
  - Provide a link to the product-specific style and terminology guide that applies to the content, if one exists.
  - The URL to the master branch staging URL for the content you want edited.

## After an edit pass is filed

- The baseline SLA for edit completion is five business days. However, the actual delivery of your edits will be based on volume of edit requests.
- The preferred edit process is to edit content after it has been published.
- Notification of start and completion of the edit is via the work item notification from TFS. So, make sure you have TFS notifications enabled (Settings>Alerts>New). In GitHub, editors use the @ syntax (@) in comments to send notifications to let you know the edit is ready for review.
- When an edit is complete, the editor notifies the author that the edit is complete and ready for review through the TFS or GitHub notification mechanisms. The author has five business days to respond to the request.
- If the author doesn't respond to the request for review within five business days, the editor signs off on the edit. At this point, the author assumes responsibility for any issues introduced during the edit pass.

## Finalize an edit

For GitHub-based content, the edits are recorded in a pull request. When all edits are agreed to, the author adds the #sign-off comment to indicate the edit is ready to be merged.

## Questions/Concerns/Feedback

Justin Chappell (justinc) – MSFT project owner and content business SME

Tami (v-tamif) – Aquent editorial service manager

# Publishing for the air-gapped cloud (AGC)

5/21/2021 • 6 minutes to read

Air-gapped cloud (AGC) environments are private environments without internet connectivity for high-security projects like JEDI. Engineering teams are expected to work with doc teams to publish content about any differences in how Microsoft services function within an AGC. That content is published as a supplement to our public content on Docs and Learn sites in AGC by using restricted, internal Azure repos rather than GitHub.

Publishing to AGC requires tenting. Documentation managers can nominate a **limited** number of contributors on a need-to-know basis. Contributors must be tented before they can request access to the AGC repos. The rest of this topic explains more about content intended for AGC, how to request access, and how to publish.

## Restricted repos

Restricted Azure repos for AGC are:

- Private
- Published to review.docs.microsoft.com but never to public docs.microsoft.com site
- Live branches included in the teleport package to the designated air-gapped environments
- English only and not localized
- A unique URL to keep both the review site and the published urls unique

Restricted repos can't include:

- **HBI info:** Information that violates HBI or security policy in aggregate
- **High-side links:** links to the Azure portal on the high side
- **Region-specific links:** Links specific to the region as the root URL is restricted information
- **Classified info:** U.S. Government classified information
- **Critical Information List:** Any information listed on the [Critical Information List](#)
- **Blog posts:** Blog posts are typically written in the first-person voice and are related to announcements and promotions. They often sound like a personal story.
- **Code and project samples:** There is no support for code and project samples in the air-gapped clouds at this time. The Docs and Learn team is working on an approach for this type of content.
- **Community spotlight/community resources:** Do not publish articles featuring community projects. Docs.microsoft.com is for technical content about how to use the product or service described from the Microsoft perspective, not about how people are using the product. That's marketing or possibly blog content. Or, let the community tell its own story in the places that community likes best!
- **White papers:** White papers are typically .pdf files hosted outside of Docs. External links outside air-gapped clouds are inaccessible.
- **Downloadable files:** Technical documents should be delivered as articles, not downloads. Other downloadable content should go to the solution developed by Docs and Learn indicated above - binaries are included in this category and are not allowed in Docs repositories.
- **Pricing:** We don't publish pricing information on Docs. Requests for pricing information should be redirected to the customer's sales team.

- **Future product plans or promises:** Do not publish statements about future product plans in technical documentation. Technical documentation should describe only what is possible in the released product. Why?
  - **It's not actionable:** Technical content should be about what customers can and should do today, not what they might be able to do some day.
  - **It erodes trust:** If a writer says a feature is coming soon and it is delayed or canceled, the statement risks eroding trust for all our technical documentation.
  - **Looks like marketing:** Talking about the future looks more like marketing material (aspirational rather than reality).
- **Legal terms and software licenses:** Legal terms and software licensing are published to webpages such as Microsoft Office 365 Legal Information and Microsoft - License Terms. No general legal terms or software licensing should be published to Docs. However, some product groups do publish more granular policies or terms to Docs, under the guidance of their CELA representative. All groups must host this content in a centralized repo and follow a special localization process. For detailed information, see Legal content.
- **SLA:** Ordinarily, SLA discussions should occur on the SLA pages for the service. As the public SLA pages are inaccessible within the air-gapped clouds, we may choose to publish AGC-specific SLA materials in the restricted repos. More information TBD.
- **Private preview documentation:** The docs.microsoft.com site does not support private preview content. Product teams need to find other channels for documentation that supports products that are available only through private preview programs. Only technical documentation for publicly available services and software can be published to docs.microsoft.com. Additionally, the following items are also disallowed on the Docs AG-specific repositories:
  - **Binaries**
  - **Scripts**
  - **Sample source code**

## When to use a restricted repository

- Non-cleared content about features and other materials that will never ship to the public and need to be given to those using our air-gapped clouds. An example would be some of the content for Azure Stack products specific to JEDI.
- Content about differences for a Microsoft service or product feature that need to be called out AGC.

## When not to use a restricted repository

- Private preview
- Third-party content

## Working with tented content

From start to finish, treat content that will be published to AGC as tented information. Before discussing tented information, make sure everyone in the discussion is tented. Don't delegate content tasks to anyone who is not tented. Engineering partners, writers, reviewers, and anyone else who works with the content must be tented. If a documentation team needs to publish to AGC but doesn't yet have a tented contributor, send a request to Air Gapped Cloud Pull Reviewers [agcpr@microsoft.com](mailto:agcpr@microsoft.com).

## Access to air-gapped cloud repos

After tenting, a contributor can request access to any of the AGC-only content repositories.

1. Go to [MyAccess](#)
2. Select 'Request Access'
3. Search for `Docs and Learn AGC` Your request will be routed to your manager for approval.

#### NOTE

For Microsoft 365 repos, please contact David Zazzo [David.Zazzo@microsoft.com](mailto:David.Zazzo@microsoft.com) before applying for access. For Power Platform repos, please contact Buckley Guderian [buckley.guderian@microsoft.com](mailto:buckley.guderian@microsoft.com) before applying for access. For Azure repos, please contact Justin Hall [Justin.Hall@microsoft.com](mailto:Justin.Hall@microsoft.com) before applying for access.

## Restricted repo names

If you are new to publishing, sign up to attend an [onboarding class](#).

There are currently two restricted repositories for Azure content.

- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/azure-docs-agc-usnat](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/azure-docs-agc-usnat)
- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/azure-docs-agc-ussec](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/azure-docs-agc-ussec)

There are currently two restricted repositories for Office content.

- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/microsoft-365-docs-agc-usnat](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/microsoft-365-docs-agc-usnat)
- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/microsoft-365-docs-agc-ussec](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/microsoft-365-docs-agc-ussec)

There are currently two restricted repositories for Power Platform / Power BI content.

- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/power-platform-docs-agc-usnat](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/power-platform-docs-agc-usnat)
- Repo URL: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/power-platform-docs-agc-ussec](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/power-platform-docs-agc-ussec)

If your business group requires a separate repo to create the right url path, your business group needs to:

- Manage the repo and comply with security policies
- Delegate someone to approve access requests

## Content management

Follow these best practices for managing content in restricted repos:

- **Folder naming:** Name folders after the service name that the content covers. For example, azure-stack-edge.
- **Index file:** Include an index file in each folder so there is a landing page specifically for the files within that folder.
- **Toc file:** Nest TOC structures so that topics that have information specific for air-gapped cloud appear at the top of the TOC, followed by public docs for the service.

## Authoring workflow

The workflow between GitHub and Azure DevOps is similar. If you're using VSCode with the Docs Authoring pack, you can still do so, however Acrolinx won't be available. For information on general steps you can adapt, see [Make changes to an article](#).

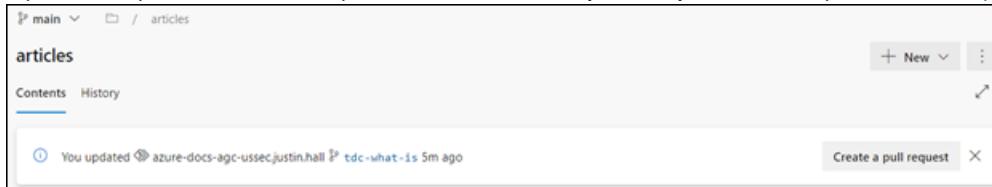
#### NOTE

For Azure repos, please contact [Joelle Faustin](#) before contributing your first pull request. Joelle is coordinating region specific updates. As noted above, for Office Microsoft 365 repos, the contact is [David Zazzo](#).

While most steps in [Make changes to an article](#) apply, these repos do not currently use the #sign-off workflow. Also of note, if you are ready for your content to be seen inside the air-gapped regions, you must push your content to the live branch. It is the **live** branch which is picked up for delivery into the regions.

## How to add an article

1. Go to the repo. For example: [https://ceapex.visualstudio.com/DefaultCollection/AGC/\\_git/azure-docs-agc-ussec](https://ceapex.visualstudio.com/DefaultCollection/AGC/_git/azure-docs-agc-ussec)
2. Clone the repo, but don't create a fork.
3. Create a working branch.
4. Edit or add a new topic.
5. in Git Bash, type: git add
6. Type: git commit -m "message"
7. Type: git push origin
8. Open the repo on Azure DevOps and click **Create a pull request**. For steps, see [Create pull request](#).



9. Add required reviewers alias *agcpr* and click **Create**.

New pull request

azure-docs-agc-ussec.justin.hall tdc-what-is into azure-docs-agc-ussec main

Overview Files 1 Commits 1

Title  
new overview

Description  
new overview

Markdown supported. Drag & drop, paste, or select files to insert.

Link work items.

A **B** *I* `</>`

new overview

Reviewers Add required reviewers

AGC Pull Request Reviewers +

Work items to link

Search work items by ID or title

Tags

Create

This screenshot shows the 'New pull request' interface in Azure DevOps. At the top, it displays the repository path: 'azure-docs-agc-ussec.justin.hall tdc-what-is into azure-docs-agc-ussec main'. Below this, there are tabs for 'Overview', 'Files 1', and 'Commits 1'. The 'Overview' tab is selected. The main area contains fields for 'Title' (containing 'new overview') and 'Description' (containing 'new overview'). Below the description is a rich text editor toolbar with icons for bold, italic, code, link, list, at symbol, hash, file, and copy. To the right of the description is a link to 'Link work items.' and a note about Markdown support. Below the editor is a preview of the content. The 'Reviewers' section shows a group named 'AGC Pull Request Reviewers' with a '+' button. The 'Work items to link' section has a search bar. The 'Tags' section is empty. At the bottom right is a 'Create' button with a dropdown arrow.

10. After the pull request is approved, select **Delete working branch** and click **Complete > Complete Merge**.
11. In order for the content to be published and pushed to the air-gapped environment, you must merge to the live branch. For new repos, an engineering PM needs to allow publishing through the Docs portal first.

# Metadata overview

5/10/2021 • 3 minutes to read

At Microsoft, we use metadata on Docs for reporting, discoverability of the content via search, and to drive aspects of the site experience. Following the guidelines in these articles will help make sure your content continues to function as expected as features change.

For reporting, metadata particularly drives the [Content & Learning reports](#). Metadata as it is applied to content can also be queried directly through Kusto. The allowlists we maintain, such as ms.prod and ms.service, can be viewed on the [Taxonomies for Docs](#) page in this guide. If you don't set metadata fully and correctly, your content will be misrepresented or underrepresented in reports, resulting in less information to help you understand and serve customers.

## NOTE

For help with reports, see the [Data and BI overview](#).

For the site experience, metadata helps drive the search API and site search behavior and generates various site features like scoped searches, code sample highlighting, and RSS feeds. If you don't set metadata fully and correctly, Docs site features will not work properly on your content, resulting in a subpar user experience.

## National vs. local level analogy

When talking about metadata governance, we use the "national" vs. "local" government analogy.

The guidance in this documentation sets "national" rules for metadata. These apply to all content on Docs, as national (federal) laws in the United States apply to the whole country. We allow and encourage people responsible for individual content areas to identify their unique requirements and document additional guidelines for their areas. These can be thought of as "state" or "local" rules for metadata.

We encourage you to designate someone in your organization to be the point person for metadata. They can create a document based on these guidelines and place it in the "Repository-specific Guidance" section of the contributor guide. However, your "local" guidance may not contradict the "national" rules we set in these documents.

When developing local guidance, you should include:

- Any attributes that are [considered optional for Docs](#) (meaning that Docs-wide systems will not assume that all content has this attribute filled out) but are required for your workflows or system connections.
  - Example: If this content is being produced for Ignite 2021, include ignite-2021 in ms.custom.
- Editorial guidelines like conventions for title suffixes.
- A metadata application strategy for national, rules-based, and direct file metadata.
- Subsets of Docs allowlists that are approved for your organization and explanations of why to choose one or the other.
  - Example: Valid `ms.prod` values are A, B, and C.
- Guidance on how to use `ms.reviewer`, if at all. The standard Docs guidance is to make it the alias of a person that reviews the content, but most teams should provide specific guidelines on choosing the right individual.
  - Example: `ms.reviewer` should be the M1 for this content area.
- Specific guidelines for using `includes` within your repo.
- Other attributes required to connect your content on Docs to F1 help systems, your individual reporting

systems, or any other crosswalks necessary between Docs and systems you use.

- Example: Include `ms.search.industry` on all content, to enable in-product help search.
- Guidance about how your team uses `ms.custom`, specifically.
- If your area has reference content, document the default values that should be set and the reasoning behind any major decisions.

## Stay up to date

New `ms.prod`, `ms.technology`, `ms.service`, and `ms.subservice` values are processed weekly. Up-to-date lists are available on the [Taxonomies for Docs](#) page. Requests for major changes to metadata are assessed by the Docs Metadata Governance Committee once a month, and the status of all requests can be seen on the [Metadata Requests board](#).

## Contact

For help with anything metadata-related, use the [Docs Support teams channel](#). This channel is the best place to get questions answered quickly and is staffed by a member of the Docs PM team and the Docs engineering team every day.

If you know you have a request to file, start by reading about [the different kinds of changes we make](#). For help with reports, see the [Data and BI overview](#).

# How to apply metadata

6/9/2021 • 7 minutes to read

Metadata can be applied to content in several ways, depending on how broadly the metadata values apply:

- Globally for an entire docset.
- For an entire folder or wildcarded path.
- For an individual article.

We recommend applying metadata at the broadest level you can, so that contributors don't have to do it manually in every file. Manually applying metadata in files is error prone: people tend to forget, or make typos, or add the wrong values. If you add a default value that is appropriate for a large set of content, contributors don't have to worry about it!

For example, if all the articles in a docset are about the same product and technology, you can set `ms.prod` and `ms.technology` at the docset level. If all the articles in a folder have the same author, you can set `author`, `ms.author`, and `manager` at the folder level. Values that are unique to each article, such as `title` and `description`, must be set at the file level.

## IMPORTANT

If metadata is set in more than one place, the lowest level wins. That is, file-level metadata overrides folder-level, and folder-level overrides global. This allows you to make exceptions if most, but not all, articles in a scope share the same value. If you set metadata for a docset or folder, make sure to communicate to contributors and PR reviewers that you have done so and document it in your local metadata guidance, so they don't inadvertently override correct metadata with different values!

## How to apply metadata to a docset

Applying metadata globally requires editing the `docfx.json` config file. This is generally restricted to repo admins and others empowered to make global changes for a docset. Consult with your repo owner before editing `docfx.json`!

1. In the `docfx.json` file for the docset, find the `globalMetadata` array.
2. Within the curly braces that define the array, add an entry for the metadata field you want to set, such as `author`.
3. Add the default value you want to set, such as "janedoe".
4. Make sure all entries in the array are separated by commas.

Here's an example `globalMetadata` section with `author` and other fields set:

```
"globalMetadata": {  
    "breadcrumb_path": "/biztalk/breadcrumb/toc.json",  
    "layout": "Conceptual",  
    "author": "janedoe"  
},
```

Now every article in the docset will have the `author` value "janedoe", unless it is overridden by a folder- or article-level value.

## How to apply metadata to a folder or path within a docset

1. In the docfx.json file for the docset, find or add a `fileMetadata` section.
2. If it doesn't already exist, add an entry for the metadata you want, such as `author`.

### IMPORTANT

If there's already an entry for the metadata field you want to define for a folder path, add your paths to the same entry. If multiple entries are added for the same metadata, all but one will be ignored.

3. Define the rule you want to use for applying the author metadata. This is usually a folder path and file name extension, such as `articles/**/*.md`.
4. Add the value, such as "johndoe", after the rule. The following example sets a default author value of "johndoe" to every Markdown file in the `articles/test` folder of a repo and all sub-folders, and a default author value of "amydoe" to every Markdown file in the `articles/test2` folder:

```
"fileMetadata": {  
    "author": {  
        "articles/test/**/*.md": "johndoe",  
        "articles/test2/**/*.md": "amydoe"  
    }  
},
```

## How to apply metadata to an article

For values that are unique to each article, specify them directly in the file's YAML front matter, also known as the *metadata block*.

Article metadata is maintained by the article author or the product content team. To apply or update metadata in an article, create a pull request in the private repository.

Here's an example of metadata applied in the YAML front matter of a Markdown article:

```
---  
title: # the article title to show on the browser tab  
description: # 115 - 145 character description to show in search results  
author: {github-id} # the author's GitHub ID - will be auto-populated if set in settings.json  
ms.author: {ms-alias} # the author's Microsoft alias (if applicable) - will be auto-populated if set in settings.json  
ms.date: {@date} # the date - will be auto-populated when template is first applied  
ms.topic: getting-started # the type of article  
---  
# Heading 1 <!-- the article title to show on the web page -->
```

Attributes are case-sensitive. Ensure that you enter them exactly as listed and use a colon and space between the attribute and value. If an attribute value includes a colon (:) or a hash (#), it must be quoted using single ('') or double ("") quotes. For example:

```
---  
title: 'Quickstart: How to use hashtags (#) to make a point on the internet'  
---  
# Heading 1 <!-- the article title to show on the web page -->
```

# Single-valued attributes vs. multi-valued attributes

The examples shown so far are all of single-valued metadata. Single-valued metadata should always use the scalar format shown - that is, the value on the same line as the attribute, separated by a colon and a space.

Some attributes, such as `ms.custom`, are allowed to have multiple values. YAML supports the following array formats:

Each value on its own line:

```
---
ms.custom:
  - high-priority
  - 'created by: mbradley'
---
# Heading 1 <!-- the article title to show on the web page -->
```

Values in a comma-separated list within brackets:

```
---
ms.custom: [high-priority, 'created by: mbradley']
---
# Heading 1 <!-- the article title to show on the web page -->
```

Like scalar values, values in arrays should be quoted if they contain a colon (:) or hash (#).

## Bulk update tools

You can file a new [sitehelp issue](#) to request help from the production team when you have a bulk update request. Visual Studio Code has a [multiline search feature](#) that's really handy for updating metadata in bulk. There are also several home-grown tools used for bulk updates of metadata:

- [Metadata Update Tool](#)
- [Dr. Move](#)
- Other tools in the [Azure-CSI-Content-Tools repo](#)
- PowerShell script to replace metadata strings in multiple files in small batches.

Copy the following code and save it in a `.ps1` file. Edit the two parameters in the code, or run it with the following switches: `.\metatadata.ps1 -contentDir .\yourfolder\ -batch 0`

This script loops over a batch of 10 Markdown files that have `title:` in the metadata. It removes the lines for `ms.suite`, `ms.workload`, and `ms.tgt_platform`, and replaces the `ms.topic: article` with `ms.topic: conceptual`. You can customize the `select-string` and `foreach-object... replace` lines with the edits you want to do.

After processing a file, it opens the file in VS Code for you to review manually before committing. Notice the changed lines are flagged with green, and the removed lines are flagged with red.

Then you can make a PR or new working branch from each batch of 10, or batch many files into one bigger PR. Run the script again for the next 10 Markdown files by incrementing the batch counter in the parameters, such as `$batch = 1`.

```

param(
    [String]$contentDir = "C:\git\azure-docs-pr\articles\<your folder>",
    [int]$batch = 0      # start with 0 for first 10 files, 1 for next 10 files, etc.
)

$files = dir $contentDir\*.md | select-string "title:" | select-object path -Unique
Write-output "Found $($files.count) files that need to be changed. Changing a max of 10 for an auto-accept PR."
$batch=$batch*10
foreach ($file in $files[$batch..($batch+9)])
{
    write-output "File: " + $file.Path
    (Get-Content $file.Path) |
    Select-String "documentationcenter: \\'\\\'" -NotMatch | # remove these lines
    Select-String "ms\.workload:" -notmatch | # remove these lines
    Select-String "ms\.tgt_pltfrm:" -notmatch | # remove these lines
    Select-String "ms\.devlang: na" -notmatch | # remove these lines where NA is used
    Select-String "manager:" -notmatch | # remove these lines
    Foreach-Object { $_ -replace "editor: oldgitalias", "editor: newgitalias" } | # find replace example
    Foreach-Object { $_ -replace "ms.topic: article", "ms.topic: conceptual" } | # find replace example
    Set-Content $file.Path
    code $file.Path
}

```

## Pull request for bulk updates

When you do a bulk update of metadata across many files, set up your pull requests for easy review. Follow these best practices:

1. Split up bulk updates into batches of 100 files or fewer, and use a separate Pull Request for each batch. Title the pull request as "Bulk metadata update:". If you are strict about scoping the changes so they are very consistent and obvious, you can submit larger pull requests. It is still a good idea to batch changes for easy review.

If you have several related pull requests, you can give the C&L Pull Request Review Team (use the [techdocprs](#) alias) a heads-up. This kind of bulk update does not require an Acrolinx score of at least 80.

If you need to divide a large pull request, see [Break up a single large pull request into smaller PRs](#).

2. Discuss with the repo admins to merge such requests directly, rather than using the #sign-off process and engaging PRMerger.

# Tagging guidelines

3/5/2021 • 9 minutes to read

These guidelines were created to help content contributors better understand what tags support across all content on Docs.Microsoft.com and make more informed choices about the tags that they apply to content.

These guidelines sit at the "federal" level of taxonomy management and tagging workflows, which means they are the most general guidelines and apply to all content on Docs.Microsoft.com. Specific areas of Docs.Microsoft.com, such as Learn or PNP, and individual content teams may want to further define their own tagging guidelines to support their content efforts. State and local guidelines should **not** conflict with the Federal guidelines so we can maintain consistency across the entire content ecosystem.

**Federal:** All content on Docs.Microsoft.com

**State:** A specific part of the site, such as Learn

**Local:** An individual content team within a "state"

## What our taxonomies support

The tags we apply to content come directly from a set of taxonomies managed by the IA team and live in a taxonomy management tool called PoolParty. PoolParty feeds the taxonomy terms and data into the other systems on the Docs.Microsoft.com platform. These tags support four areas of user experience and content management: findability, discoverability, content strategy, and quality assurance.

### Findability

One of the biggest things our taxonomy-supported tags supports is **findability**: how easy it is for users (both internal to Microsoft and our customers) to find a known-item within our vast content ecosystem. Examples of a known-item search might include API reference documentation for a specific product, or a troubleshooting guide for using a specific programming language in a product suite.

Our taxonomies support this by supplying the filters on **browse pages** and **on-site search**, largely used by end users. But this is also important for internal content creators, as well, so that our content teams can **find content across repos** and **curate collections** in the future.

### Discoverability

The way we apply tags from our taxonomies to our content also heavily supports **discoverability**: a user's ability to understand and use the relationships between different pieces of content and concepts to find more information relevant to their problem, task, or interests. This is most often exposed through **content recommendations** and showing **related content** across Docs.Microsoft.com, allowing us to enable experiences like showing related documentation from Docs on relevant Learning Paths within Learn. Discoverability also supports our content creators at Microsoft to **identify content for reuse** without having to already know the content exists.

### Content strategy

How we tag our content directly impacts the ways teams can create and support meaningful content strategies: the creation, delivery, and governance of useful, usable content. The tags applied to our content directly impact **reporting**, how we can actually understand what is happening with our existing content and how our users interact with it. We can also better understand the **gaps in coverage** within a content set, identifying how many content pieces are tagged with specific terms and not others, which in turn helps us in **content planning** efforts, so we can fill in gaps and better meet customer needs.

## Quality assurance

Our tags also play an important role on a system level. When content is approved for publishing, \*\* \*\* our tags provide **metadata validation** to make sure the content we're publishing meets established standards. Some of our tags also **enable site functionality** that relies on valid taxonomy-approved values to render correctly, such as our code highlighting capabilities on Docs documentation.

## Taxonomy governance

The IA team manages the day-to-day governance of the DevRel taxonomies. This team does weekly updates that may include the addition of new values or the management of existing values. New terms may be requested by emailing [devrellIA@service.microsoft.com](mailto:devrellIA@service.microsoft.com).

To request the creation of new taxonomies, the IA team prioritizes taxonomy creation based on feature prioritization. Taxonomy creation can be a complex process, so we are unable to create and implement new taxonomies without significant work. To request a feature to create a new taxonomy, please fill out the [DevRel Feature Request form](#).

## General tagging guidelines

### Working with hierarchies

Assume that tagging with a child term (L2 or deeper) will automatically assign the parent term(s).

- Ex: Tagging a piece of content about Microsoft Excel doesn't also need a tag for Microsoft Office. Since Microsoft Office is the *parent* term to Microsoft Excel, the system can automatically apply the parent tag appropriately.

### Number of tags

Generally, **no more than five tags per taxonomy** should be used on a single piece of content. In some cases, not all taxonomies will apply to your content, so no tags will be applied from those taxonomies. Applied tags should be specific, so choose the **most** specific tag(s) for your content. Too many tags dilute the usefulness of the tags and users have difficulty understanding what they should actually expect from the content.

- Ex: Tagging a piece of content about Microsoft Office should not also include child terms like Microsoft Excel, Microsoft PowerPoint, and any other product in the Office suite. If the content is not directly concerned with Microsoft Excel, adding that product tag is not useful for people searching for specific information about either Excel or Office in general.

### Specificity of tags

Related to the number of tags applied to content, those tags should also be specific, only describing the products/concepts that the content is directly concerned with.

### Working with tag recommendations

As of this document, only the Product taxonomy has been used in machine learning-based auto-tagging work. To scale tagging for existing content and lighten the mental load on authors to choose the best tags, we plan to roll out this auto-tagging capability for all taxonomies and incorporate it into the authoring workflow.

The auto-tagging done by PoolParty is currently set to apply a maximum of **five product tags** per piece of content, though many pieces of content receive fewer tags. This is also a good general rule to follow when selecting tags manually.

### Other considerations

- If I wanted this content to appear on the first page of Google, would this tag be a useful keyword? Is it specific enough?
- What are the top three keywords you would use to describe this content?

- Some content may be about more than one product, and it's fine to tag with more than one product term when needed. But be judicious and conservative with how many product tags you apply.
- If you find yourself thinking "it could be about this, too...", probably not a useful tag.

**Your motto for tagging your content: Be judicious. When in doubt, don't tag**

## Specific guidelines for each taxonomy

### Product

The [Product taxonomy](#) covers products and services created, maintained, and sold by Microsoft. We don't include third-party products or services, nor can we maintain terms specific to product features. These terms are currently used across Learn, Architecture, and Samples content. This is our most frequently used taxonomy. We are in the process of setting up the process for tagging all Docs.Microsoft.com content with "Product" terms.

### Tagging Rules

- Minimum product tags: one (required). Maximum product tags: five.
- Only Microsoft products are tagged.
- If content addresses **what** a product does, **why** it should be used, or **how** it is used, then tag content with that product term.
- Do not tag related products that are not directly addressed in the content.
- If content is relevant to:
  - One child term, tag with child term
  - Some child terms, tag with each child term
  - All child terms, tag with parent term (no child terms)
  - Parent term, tag with parent term only

**Examples** Each of these examples are a fit for the Kubernetes product tag because they address different aspects of the product:

- **What:** This article answers "what" AKS is: <https://docs.microsoft.com/azure/aks/intro-kubernetes>
- **Why:** This article answers the "why" of choosing AKS by discussing choices and tradeoffs: <https://docs.microsoft.com/azure/architecture/reference-architectures/containers/aks/secure-baseline-aks>
- **How:** This article answers "how" to deploy a Kubernetes service: <https://docs.microsoft.com/azure/aks/kubernetes-walkthrough-rm-template>

### DevLang

The [DevLang taxonomy](#) covers development languages used by, and compatible with, Microsoft products and services. Terms in this taxonomy include languages developed and maintained by both Microsoft and non-Microsoft entities. Platforms and frameworks are not currently included in this taxonomy. Currently used on Samples and AAC content to power browse filters. Likely to be extended across Docs when capability exists.

### Tagging Rules

- If content includes an example of code using a specific programming language, tag content with that DevLang term.
- If conceptual content is about a specific programming language, tag with that DevLang term (programming language should be a primary topic within the content itself).
- If a programming language is not directly mentioned or addressed, do not tag with that DevLang term.
- If the content notes that an exercise could be done using a different programming language but no detail is provided about how, do not tag with that alternative programming language DevLang term.

### Examples

- This article should only be tagged with the Python DevLang tag: <https://docs.microsoft.com/azure/app-service/tutorial-python-postgresql-app?tabs=bash%2Cclone>
- This module mentions Java, but is about how to use C#. Only the C# DevLang term should be applied: <https://docs.microsoft.com/learn/modules/introduction-to-xamarin-android/2-what-is-xamarin-android>

## Role

Our content is sometimes created with a specific general job role as the focus, especially in instructional content. The [Role taxonomy](#) defines those broad job roles we create content for. This taxonomy doesn't define specific industries or broader audiences to whom our content may be relevant. Currently used on Learn and Certifications content to power browse filters.

### Tagging Rules

- Tag content with a Role only when it is specifically relevant to someone working in/performing that role. Content should correspond specifically to the Role(s) being tagged.
- Role tags should be limited; Role is an easy concept to overtag, which then reduces the effectiveness of the tag(s) applied.
- For content that is applicable across multiple roles, do not tag all roles that could possibly match. Tag only the primary role that would be responsible for that function.

## Examples

- This module should be tagged with Data Scientist: <https://docs.microsoft.com/learn/modules/train-evaluate-regression-models/>
- This module should be tagged with Data Analyst: <https://docs.microsoft.com/learn/modules/data-analytics-microsoft/>

## Level

For instructional content, the [Level taxonomy](#) defines the general degree of experience the user should have to understand the content. Currently used **only** on Learn and Certifications content.

### Tagging Rules

- Select the most appropriate Level tag for your content:
  - **Beginner** - Broad topics that are foundational to other topics.
  - **Intermediate** – Goes deeper on a given topic and assumes baseline knowledge and verbiage.
  - **Advanced** - specialized content, foundational knowledge on topic is assumed

## Examples

- This is an example of **Beginner** content: <https://docs.microsoft.com/azure/architecture/guide/architecture-styles/microservices>
- This is an example of **Intermediate** content that assumes an understanding of architecture: <https://docs.microsoft.com/azure/architecture/reference-architectures/containers/aks-microservices/aks-microservices>
- This is an example of **Advanced** content that is unique to Data Scientists: <https://docs.microsoft.com/learn/modules/train-evaluate-regression-models/>

## Certification Type

The [Certification Type](#) taxonomy defines the certification type specifically for certifications and exam content. This taxonomy doesn't define the level of skill or expertise reached by earning a certification. Currently only used on certifications and exam content and powers filter functionality in browse experiences.

### Tagging Rules

- Select the most appropriate Certification Type tag for your content if appropriate and already approved:

- **Fundamentals** - certification type class for foundational knowledge of Microsoft products (such as Microsoft Power Platform)
- **Role-based** - certification type class for roles associated with a Microsoft product (such as Azure Administrator)
- **Specialty** - certification type class for solution areas (such as Data & AI)
- **MCE** - certification type class for Microsoft Certified Educator
- **MOS** - certification type class for Microsoft Office Specialist
- **MCSA** - certification type class for Microsoft Certified Solutions Associate
- **MCSD** - certification type class for Microsoft Certified Solutions Developer
- **MCSE** - certification type class for Microsoft Certified Solutions Expert
- **MTA** - certification type class for Microsoft Technology Associate

#### **Examples**

- Certification tags should only be applied with approval and/or direction of the appropriate Certification team.

# Metadata attributes

6/16/2021 • 15 minutes to read

This article describes required and common author-added metadata for different types of Docs content. Author-added metadata is metadata a Docs author adds to English content. Authors can add metadata to either the YAML front matter of a Markdown file, or in the `docfx.json` file for a docset. For more information, see [How to apply metadata](#).

## IMPORTANT

This article doesn't list all metadata that might be present in an article. For example, an author may add some additional metadata as part of the [localization process](#). Content teams may add other custom metadata to support their own workflows. Docs Build validates required metadata to make sure it's present and the values are valid, but doesn't validate against custom metadata unless it's specifically reserved for use by Docs. Metadata that isn't recognized by Docs is ignored by build validation.

## Required metadata

The Docs Governance Committee requires the metadata attributes in the table below. They're required unless the committee granted your team an explicit exception. One possible justification for granting an exception to a specific repository could be that the repo isn't involved in Docs content reporting. If you don't have all the following metadata in your article, one or more of the following problems will occur:

- You'll get build Warnings on your articles, potentially blocking merging and publication.
- Content reporting won't be correct.
- GitHub issues might not route appropriately.
- RSS feeds or site search might not show your content as expected.
- Your articles might perform worse for search engines.

When you attribute your content correctly, site features will perform as expected and you'll get to make the most of the content reports.

Required metadata attributes are:

FIELD	VALUE	WHY?
<code>author</code>	The author's GitHub ID.	Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, GitHub automation might notify the author of activity involving the file.
<code>description</code>	A summary of the content. 75-300 characters.	Used in site search. Sometimes used on a search engine results page for improved SEO.

FIELD	VALUE	WHY?
<code>ms.author</code>	The author's Microsoft alias, <i>without</i> "@microsoft.com".	<p>Identifies the article's owner. The owner is responsible for decisions about the content of the article, and for the article's reporting and BI.</p> <p>Can be an individual employee or an approved Microsoft group alias. To use a group alias, you must first have it approved and added to the <a href="#">ms.author allowlist</a>.</p> <p>The <code>ms.author</code> value can't be a user-created friendly email name, like <code>Jonathan.Dough</code>. The value must be the shorter email name that the company assigned to the person.</p> <p>Vendor (v-dash) aliases aren't recommended for <code>ms.author</code> because frequent turnover results in invalid values. When the ownership of the article changes, you should update this alias value promptly in the article.</p> <p>In general, <code>ms.author</code> should refer to the same person as the <code>author</code> value, unless <code>ms.author</code> is a group alias, in which case you'll need to assign an employee with a GitHub account as the owner in the <code>author</code> field.</p>
<code>ms.date</code>	A date in the format MM/DD/YYYY.	Displayed on the published page to indicate the last time the article was substantially edited or guaranteed fresh. The date is entered without time and is interpreted as 0:00 and in the UTC time zone. The date displayed to users is converted to their time zone. If this field is missing, the date of the last commit is displayed instead, which may be incorrect for freshness.
<code>ms.service</code> <i>or</i> <code>ms.prod</code>	The service or product. Use one or the other, never both.	<p>Used for issue triage and reporting.</p> <p>Generally, use <code>ms.service</code> for cloud applications and use <code>ms.prod</code> for on-premises servers and applications. If you need help with this entry or have any questions, contact <a href="#">Docs Metadata Management</a>.</p> <p>Depending on the product or service, <code>ms.prod</code> might take <code>ms.technology</code> as a child value and <code>ms.service</code> might take <code>ms.subservice</code>. For available value pairs, see the <a href="#">ms.prod</a> and <a href="#">ms.service</a> allowlists.</p>

FIELD	VALUE	WHY?
<code>ms.topic</code>	The type of content.	<p>See the <a href="#">quick reference</a> section below for information about common topic types. See the <a href="#">ms.topic allowlist</a> for the complete list of valid values.</p> <p>Usually one of the following values:</p> <pre>article, conceptual, contributor-guide, interactive-tutorial, overview, quickstart, reference, sample, tutorial.</pre>
<code>title</code>	The page title.	<p>It's the most important metadata for SEO. For more information, see <a href="#">SEO: Tips for writing titles</a>.</p>

#### IMPORTANT

For compliance, .NET reference and unmanaged reference like Win32 and COM require APIScan metadata. For more information, see [APIScan](#).

## ms.topic quick reference

To make clean comparisons between article types, follow this guidance below when assigning a value to

`ms.topic`.

#### IMPORTANT

For the full list of valid values, see the [ms.topic allowlist](#).

See the map of topic types to docs model for our [Content Performance Reports & dashboard](#).

MS.TOPIC VALUE	DESCRIPTION
<code>archived</code>	Use in archived files only. Archived content stays on the site. A banner announces it as archived documentation.
<code>article</code>	<p>Content that provides utility but was not planned by a core writing team. Typically, the author references this file in a separate ToC or docset.</p> <p><b>Note:</b> In the past, <code>article</code> was used for "any general article". The new guidance is to use <code>conceptual</code> in that case.</p>

MS.TOPIC VALUE	DESCRIPTION
<code>conceptual</code>	<p>Any content linked in a ToC that <i>isn't</i> marked in metadata as a:</p> <ul style="list-style-type: none"> <li>• Quickstart</li> <li>• Tutorial</li> <li>• Sample</li> <li>• Reference</li> <li>• Overview</li> <li>• <i>any other article type in this table</i></li> </ul>
	<p>We consider this article type as the <b>default</b> article type. If you're uncertain which type to choose, use this value.</p>
<code>contributor-guide</code>	<p>Any content in a content contributor guide.</p>
<code>end-user-help</code>	<p>Content aimed at end users that provides resolution on a specific issue.</p>
<code>error-reference</code>	<p>Reference article for common Visual Studio and .NET error codes. Despite their necessity from an SEO perspective, they often underperform. It's useful to slice them from reporting views for analysis.</p>
<code>github-samples</code>	<p>Used in <a href="#">Docs Samples</a> autogenerated files only. Don't manually add to markdown content.</p>
<code>how-to</code>	<p>Any procedural content that isn't a Quickstart or Tutorial. Located under the <b>How-to guides</b> node of the ToC.</p>
<code>hub-page</code>	<p>Use in hub page files only.</p>
<code>include</code>	<p>Use in <code>![INCLUDE]</code> files only. It's required so Docs Build, validation, and other tools to work correctly.</p>
<code>interactive-tutorial</code>	<p>Any interactive content, like TripleCrown, Hands-On Labs, or articles with Content &amp; Learning-supported interactive learning elements.</p>
<code>kb-support</code>	<p>Use <i>only</i> for files owned by the CSS support organization, regardless of which repository stores the articles. Knowledge Base (KB) articles use this value. They and don't use the <code>troubleshooting</code> value.</p>
<code>landing-page</code>	<p>Use <i>only</i> in landing page files.</p>
<code>language-reference</code>	<p>Set this value globally in <code>docfx.json</code> to identify a docset as autogenerated unmanaged API reference (Java, Node.js, Python, Azure CLI) or as authored API ref articles (C++).</p>
<code>managed-reference</code>	<p>Use in <code>docfx.json</code> config to add to autogenerated managed API ref articles (.NET SDK, PowerShell, .NET languages). For authored articles, use only for authored managed language syntax articles.</p>

MS.TOPIC VALUE	DESCRIPTION
<code>overview</code>	Overview or "User Guide" articles. These articles live under an <b>Overview</b> node in a ToC, or are high-level conceptual articles in a "New User" or "Onboarding" guide.
<code>quickstart</code>	Articles placed under a <b>Quickstart</b> node in a ToC. These articles <i>must</i> follow the specific guidelines and template for a Quickstart document.
<code>reference</code>	Any authored (read: <i>not</i> autogenerated) content placed under a <b>Reference</b> node in a ToC. For autogenerated API reference, see <code>language-reference</code> and <code>managed-reference</code> . For the specific cases of Visual Studio and .NET error reference pages, see <code>error-reference</code> .
<code>retired</code>	Use in retired files only. We remove retired content from the site and upload it as a PDF to the Download Center. A landing page lists all of the retired content.
<code>sample</code>	Any authored (read: <i>not</i> autogenerated) content placed under a <b>Samples</b> or <b>Examples</b> node in a ToC.
<code>tutorial</code>	Articles placed under a <b>Tutorials</b> node in a ToC. These articles <i>must</i> follow the specific guidelines and template for a Tutorial document.
<code>troubleshooting</code>	<p>Articles that help users solve a common issue. Use this value to identify FAQs, or to identify specific error or troubleshooting articles, so you can report on them.</p> <p>Knowledge Base (KB) articles owned by the CSS support organization don't use this value. They use <code>kb-support</code> instead.</p>

## Optional metadata

For further SEO or reporting needs, you can also add attributes in the table below. Regular Docs platform features and reporting won't assume you have any of these attributes filled out, but your team's workflows or connections to outside systems might.

FIELD	VALUE	WHY?
<code>ms.collection</code>	The collection name.	An attribute requested by the MAX org (Office). Authors can use it to tag and track subsets of content.

FIELD	VALUE	WHY?
<code>ms.custom</code>	<p><b>For writer or team use only.</b></p> <p>Commonly used for tracking specific docs or sets of content in telemetry tools. It's a single string value, and it's up to the consuming tool to parse it.</p> <p>Example:</p> <pre>ms.custom: "experiment1, content_reporting, all_uwp_docs, CI_Id=101022"</pre> <p><b>Character limit: The maximum string value length is 125 characters.</b></p>	<p><code>ms.custom</code> is a custom field that writers can use to track special projects or a subset of content. The Developer Experience docs team uses <code>devx-track-*</code> to track article groupings across services and repositories. Contact <a href="#">Barbara Kess</a> if you have any questions.</p>
<code>ms.reviewer</code>	The Microsoft alias of a person that reviews the content.	Most teams should provide specific guidelines on choosing the right individual in their local metadata guidance.
<code>ms.subservice</code>	The more granular service to which the content belongs. Only use <code>ms.subservice</code> if you're also using <code>ms.service</code>	<p><code>ms.subservice</code> by itself isn't valid metadata. The author must associate it with a parent <code>ms.service</code> value. This attribute is a way to drill down further in reporting for a given <code>ms.service</code>. See the <a href="#">ms.service allowlist</a> for valid pairs.</p>
<code>ms.technology</code>	The technology to which the content belongs. Only use <code>ms.technology</code> if you're also using <code>ms.prod</code> .	<p><code>ms.technology</code> by itself isn't valid metadata. The author must associate it with a parent <code>ms.prod</code> value. This attribute is a way to drill down further in reporting for a given <code>ms.prod</code>. In <b>Exchange</b>, for example, you could have</p> <pre>ms.prod: exchange-server-itpro</pre> <p>that applies to all evergreen Exchange Server articles, and beneath that an</p> <pre>ms.technology: high-availability</pre> <p>value that would apply to all Exchange Server articles that are about high availability in an on-premises environment. It's up to a writing team and its manager to evaluate their reporting needs and determine how granular they'd like to get with their reporting. See the <a href="#">ms.prod allowlist</a> for valid pairs.</p>

FIELD	VALUE	WHY?
<code>ROBOTS</code>	<code>NOINDEX</code> , <code>UNFOLLOW</code>	<p>Use <code>ROBOTS</code> in your metadata section to prevent the build and publishing process from showing content on search pages. When you want to use <code>ROBOTS</code> (and yes, it's all capitalized, even though other metadata tags aren't):</p> <ul style="list-style-type: none"> <li>- Add <code>ROBOTS: NOINDEX</code> to your metadata section.</li> <li>- <code>NOINDEX</code> causes the asset to not show up in search results.</li> <li>- Use <code>NOFOLLOW</code> only when you archive an entire content set. Avoid using it in individual articles you retire; it causes dead-ends for web crawlers and can hurt SEO.</li> <li>- Customers who have a direct link to assets that have <code>NOINDEX</code> and <code>NOFOLLOW</code> can still view the content, but they can't find that content using search.</li> <li>- As a best practice, don't use <code>ROBOTS</code> without a value like this <code>ROBOTS: .</code></li> </ul>
<code>contributors_to_exclude</code>	A list of GitHub aliases for contributors that you don't want displayed as contributors on the published page.	Add this metadata as global metadata in the <code>docfx.json</code> file.
<code>ms.localizationpriority</code>	Indicate a need for human translation (as opposed to machine translation) by assigning a value of <code>high</code> . A value of <code>medium</code> means that the content continues to be machine translated.	Only use this metadata tag with flat structure (no subfolders) and MVC model repos to designate which files are in need of human translation.
<code>social_image_url</code>	Specifies an image URL that's used as the source for the open graph image ( <code>og:image</code> ) metadata on the page. It should be an absolute URL on docs, including the locale (so starting with <code>https://docs.microsoft.com/en-us/</code> ) and pointing at an image file in JPG, GIF, or PNG formats (generally a PNG would be appropriate for a logo). A square image tends to work well on social media platforms.	<p>Many types of page crawlers will pick up this value and use it when sharing an article on Twitter, Facebook, and other sites. This value will <i>only</i> be output on the page if you've also specified a <code>description</code> value (listed in the <a href="#">required metadata section</a> earlier in this article).</p> <p>If this metadata isn't specified, and the author includes a <code>description</code>, the site will automatically output the Microsoft logo as the open graph image value.</p>
<code>no-loc</code>	A list of words in the article that the Localization team shouldn't translate (localized). For more information, see <a href="#">Identify content to NOT be localized</a> .	

Docs supports many more metadata attributes, though they aren't all commonly used by content authors. Contact [Docs Metadata Management](#) if you have other needs.

# Metadata for archive

We mark content that we migrated as part of the MSDN and TechNet shutdown as archived or retired.

## IMPORTANT

This section only applies to content that authors migrated to the Docs Previous Versions site. It does not apply to regular docs.microsoft.com content that teams may have decided to informally retire.

Archived and retired content metadata aren't entered by the author but instead are injected in the extraction tool config file and is populated automatically in the MD files as part of one of the extraction steps.

## NOTE

After extraction, if for whatever reason you need to update these metadata, the repo owner should be able to do that in `docfx.json` file for global or individual MD file on a topic by topic basis.

The following metadata is available for archived or retired content:

FIELD	VALUE	WHY?
APIScan metadata	Migrated from source content.	Indicates to the APIScan tool that an API is documented. It's required for compliance for .NET and unmanaged reference like Win32 and COM. For more information, see <a href="#">APIScan</a> . If source content has APIScan metadata, the author <i>must</i> migrate it to Archive.
F1 metadata Microsoft.Help.F1	If current MTPS content has F1 metadata, we automatically extract that in extracted MD under YAML	Allows a product UI to link directly to the article as Help.
<code>is_archived</code>	<code>true</code>	Shows this disclaimer on the published page: "We're no longer updating this content regularly. Check the Microsoft Product Lifecycle for more information about how this product, service, or technology is supported."
<code>is_retired</code>	<code>true</code>	Shows this disclaimer on the published page: "This content has been retired and may not be updated in the future. The product, service, or technology mentioned in this content is no longer supported."
<code>ms.author</code>	<code>Archiveddocs</code>	Indicates that there's no active author because the content isn't maintained.
<code>ms.date</code>	The extraction tool will extract the Last updated date from MTPS as the <code>ms.date</code> value. Make sure content has an <code>ms.date</code> value and <i>isn't</i> from <code>updated_at</code> metadata.	The date when the author last reviewed the content and confirmed its accuracy and relevance.

FIELD	VALUE	WHY?
<code>ms.prod</code>	Upon creating the repo, the Engagement team will provide the string value into extraction tool config file after discussing with the Product team.	
<code>ms.topic</code>	<code>archived</code> or <code>retired</code>	Indicates the status of the content.
<code>ms.translationtype</code>	<code>MT</code>	Shows a slightly different disclaimer to inform user content is machine translated.
<code>ROBOTS</code>	<code>NOINDEX</code> , <code>NOFOLLOW</code>	Content shouldn't be discoverable in Search so it's marked <code>NOINDEX</code> . Add <code>NOFOLLOW</code> only when you've archived an entire content set.
<code>title</code>	Extracted from article H1.	The title of the article.

Example from the MSDN Technet Extraction tool config file:

```
"GlobalMetadatas": [
    {
        "Key": "ROBOTS", "Value": "NOINDEX,NOFOLLOW"
    },
    {
        "Key": "is_archived", "Value": true
    },
    {
        "Key": "uhfHeaderId", "Value": "MSDocsHeader-Archive"
    },
    {
        "Key": "extendBreadcrumb", "Value": true
    },
    {
        "Key": "ms.author", "Value": "Archiveddocs"
    },
    {
        "Key": "ms.prod", "Value": "dynamicscrm-2011"
    },
    {
        "Key": "ms.topic", "Value": "Archived"
    }
]
```

## Metadata for included files

An *included file*, or *include*, is a Markdown file you can embed in another Markdown file. It's useful for shared content, like a note or warning you want to include in multiple files or a block of information that would be useful to update once and publish out to multiple locations.

The following metadata should be present on includes for all content sets.

FIELD	VALUE	WHY?
<code>author</code>	The author's GitHub ID.	Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, GitHub automation might notify the author of activity involving the file.
<code>ms.author</code>	The author's Microsoft alias.	Identifies the author by Microsoft alias. Used for reporting and other Microsoft-internal contact.
<code>ms.date</code>	A date in MM/DD/YYYY format.	Indicates the last time the file was substantially edited or guaranteed fresh.
<code>ms.service</code> (and <code>ms.subservice</code> , if used) OR <code>ms.prod</code> (and <code>ms.technology</code> , if used)	The service or product the included file applies to.	Used for issue triage and reporting.
<code>ms.topic</code>	<i>Always</i> <code>include</code> .	Identifies the file as an included file. Enables the build and publish process to run different logic on these files than on general Markdown articles. Different validation rules might apply.

Example:

```
---
author: meganbradley
ms.service: app-insights
ms.topic: include
ms.date: 05/21/2018
ms.author: mbradley
---
... include content ...
```

## Metadata for reference content

See the Onboarding Guide for [information about API reference metadata](#).

## Where did that tag go? It's not listed here

Some metadata tags are no longer required, or they're used on a team-by-team basis. That's why they aren't included on the official supported list in this page. Typically, you can remove those tags from your files and global settings. It's better to remove the tag than to leave it blank. Don't use `na` or another generic placeholder. For example:

### Manager tag

The `manager` field is commonly used in reports for rollup measurements, or when there's a process exception needing manager approval, or when there's a coordinated release that requires manager sign off to publish. In most areas of the Docs business, the content manager information is derived using a lookup to a list based on the article's `ms.service` or `ms.product` tag.

Follow the link to the **Detail** view from the `ms.prod` or `ms.service` list at [Metadata taxonomies](#) to find the

corresponding Manager value for each row there. You typically don't need to set the manager in markdown files or *docfx.json* globally, since the lookup is used. To update the manager for a certain product, technology, service, or subservice, use the [metadata changes](#) steps to update the ContentManager column.

## Other tags

Most other reference metadata is part of the reference generation process. If you have questions, post to the [Docs Support Teams channel](#) for help!

## Metadata validation

Docs Build validates required metadata and values, and the values of some optional metadata. For more information, see [Custom validation in Docs Build](#).

# APIScan

5/27/2021 • 2 minutes to read

APIScan is a tool that product teams at Microsoft run on their shipping binaries. The goal is to verify that all APIs called in Microsoft high-volume products (HVPs) are publicly documented. The documentation is a compliance requirement. It's designed to make sure Microsoft doesn't have an undue advantage over third-party developers using Microsoft products. For example, any Office API called by SQL Server must be publicly documented.

For APIScan to work, certain metadata must be set on reference articles. For example, include metadata such as the name of the API and the assembly that includes the API. The metadata is published on docs.microsoft.com, and is extracted into the APIScan database. The APIScan tool checks the database against API calls in HVPs. It checks against the API calls to verify that all called APIs are documented.

## Configuring the docset for APIScan

If you don't have rights to configure the docset or if you need help, enter a request in [SiteHelp portal](#).

### Configuring APIScan for managed reference docsets

For managed reference, required APIScan metadata is generated automatically based on information in the documented assemblies. You just have to turn it on, as follows:

1. To enable APIScan for a docset, set the **API Scan** to `True` in the [OPS self-service portal](#). The default value of the option is currently `False`, which means it skips API Scan by default.
2. As a best practice, provide the product family of the docset in [OPS self-service portal](#). The value is set to `product_family` metadata.

## Step 2 of 3: Docset information

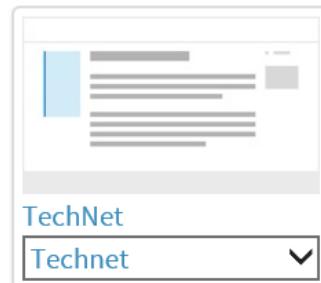
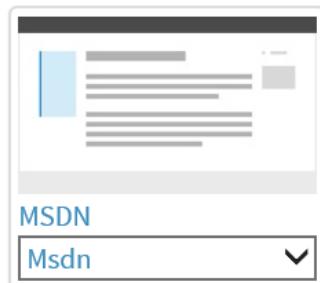
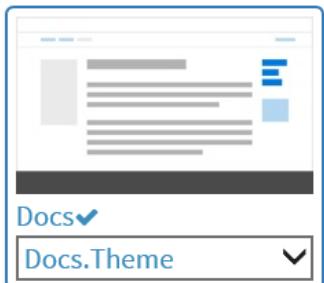
X

### Docset Name ?

▼

### Docset Folder ?

### Site/Themes ?



### Base Url ?

 docset1

### Tenant ?

▼

'Tenant' is the team/organization your documents belong to.  
Choose 'Test' if this docset is for testing purposes.

### Product Family ?

▼

Public Contributors

 API Scan

[Hide More Settings](#)

[Cancel](#)

[Previous](#)

[Next](#)

The mref build system applies the following metadata:

- **topic\_type**: The apiref. This value is the switch that turns on APIScan for an article.
- **api\_type**: The type of assembly.
- **api\_name**: The name of the API.
- **api\_location**: The DLL(s) that define the API.

## Configuring APIScan for non-managed reference

For non-managed reference, such as COM, writers must add APIScan metadata manually. Or, writers migrate it from a previous system such as CAPS. The APIScan metadata goes with other metadata in the YAML header. You can also apply it globally in `docfx.json` for values that apply to an entire docset.

```
1 ->
2   topic_type:
3     - apiref
4   api_name:
5     - sample.get
6     - sample.set
7   api_location:
8     - sample.dll
9   api_type:
10    - managed
11 --->
12
13 # Sample API 1
14
15 Test reference document
16
```

- `product` : A multi-value, specified as a global metadata in `docfx.json`. As such, all the documents in this DocSet will have same value of `product` metadata.
- `topic_type` : A multi-value and must have `apiref` as one of its values. Otherwise, this document will be skipped for API Scan.
- `api_name` : A multi-value with at least one value is specified.
- `api_type` : A multi-value.
- `api_location` : A multi-value.

2. In the `doc.fx` file, add global metadata `product` so it applies to all the docset:



1 contributor

33 lines (32 sloc) | 499 Bytes

```
1  {
2      "build": {
3          "content": [
4              {
5                  "files": [
6                      "**/*.md"
7                  ],
8                  "exclude": [
9                      "**/obj/**"
10                 ]
11             }
12         ],
13         "resource": [
14             {
15                 "files": [
16                     "**/*.png",
17                     "**/*.jpg"
18                 ],
19                 "exclude": [
20                     "**/obj/**"
21                 ]
22             }
23         ],
24         "globalMetadata": {
25             "product": ["SDK10", "Sample"]
26         },
27         "template": [
28             "op.html"
29         ],
30         "dest": "SampleAPI/v1"
31     }
32 }
```

# How to request metadata changes

6/16/2021 • 5 minutes to read

Metadata on Docs is governed by the [Metadata Governance Committee](#). As a Docs partner or content creator, you can request that we support new values, attributes, or taxonomies, or that we handle current values, attributes or taxonomies differently. The governance committee meets monthly to review requests.

You can see the changes currently being processed on the [metadata requests board](#).

## Request new terms for an existing taxonomy/allowlist

Docs currently maintains allowlists for devlang, ms.author approved DLs, ms.topic, ms.devlang, ms.prod, ms.technology, ms.service, and ms.subservice. Any other attributes can support any values.

### Request a new value for ms.service/ms.subservice or ms.prod/ms.technology

To request a new single metadata value, fill out [the form for new ms.service/subservice values](#) or [the form for new ms.prod/technology values](#). Requests submitted by end-of-day Wednesday will be processed on Thursday and live in all systems by Friday. Most teams may publish content without new values being live in the database, but that data will not show up in reports until the database is updated. Occasionally requests aren't processed within seven days, usually for the following reasons:

- **Missing information** - Requests that have missing elements often require back and forth discussion. Requests that aren't complete by the cut-off that week can't be added. Use the forms when it's practical and fill in all the required information. In most cases, we literally cannot process a request without all of it.
- **Incorrect information** - Double check that you have copied/pasted and spelled everything correctly. Frequently, it's clear that the request is incorrect, but unclear what it should be, and it causes delays.
- **Requests submitted as bugs to CGA** - Do not submit new values or changes to existing values (like changing IsContentDashboard) to CGA. We try to route them appropriately, but it's more difficult and may cause collisions.

Generally, ms.service is used for cloud services and ms.prod is used for on-prem products. If you have questions, contact [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

### To request changes or bulk updates to ms.service and ms.prod allowlists

To make changes to existing values or request more than five new values:

1. Send an email to [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com) indicating which allowlist you want to update.
2. The metadata management team will generate a spreadsheet of existing values for you to update.
3. Update any rows that you want changed (leaving the service or prod slug unchanged).
4. Delete any rows you're not changing, just to make it clearer what's changed and hasn't.

#### NOTE

We don't have the technical ability to change slug values in the database, so if you want to change the slug, add a new row with the new slug and all its associated columns and put a note by the one it should be replacing, so we can know to deprecate it.

5. Send the spreadsheet back to [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

We typically need the following information to process your request:

- Services
  - Pillar: The most general grouping for reporting. If you don't know, review the available values in the spreadsheet we send you.
  - Service area: A more specific grouping than pillar. If you don't know, review the available values in the spreadsheet we send you.
  - Service: The friendly name for the service. This field allows spaces, upper case, and lower case. It doesn't accept punctuation.
  - ms.service: The slug value, or what goes after "ms.service" in the Markdown file. Lower case only, no spaces, hyphens are allowed.
  - Sub-service: The friendly name for a more granular version of a service.
  - ms.subservice: The slug value for a more granular version of a service. It has to be associated with a service value.
  - Manager: The content manager. This field should be a Microsoft alias.
  - isDashboard: true or false. If you don't want the data to appear in the Developer Relations/Content & Learning reports, this field should be set to false. For more information, see [Report and interpret content engagement metrics](#).
  - State: Possible values are Active and Obsolete. Only change this if you would no longer like to use a certain value. Rows that are marked "Obsolete" will be omitted from reports and the allowlist display.
- Products
  - Pillar: Same as for service
  - Product: This is the friendly name for product.
  - ms.prod: The slug value, or what goes after "ms.prod" in the markdown file. Lower case only, no spaces, hyphens are allowed.
  - Technology: The friendly name for a more granular version of a product.
  - ms.technology: The slug value for a more granular version of a product. It has to be associated with a product value.
  - Manager: Same as for service
  - IsDashboard: Same as for service
  - State: Same as for service

**TIP**

Allow two Thursdays between a large request and any important deadlines that the metadata should be live for. More than 20 changes at a time often take additional time to reconcile, and will push over into the next week.

#### **Request a new DL value for ms.author**

The `ms.author` attribute value must be an individual Microsoft alias or an approved DL. To request a new approved DL, email [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

#### **Request a new value for ms.topic**

Updates to the ms.topic allowlist are rare, so there's no form available. To request a new ms.topic value, email [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

#### **Request a new value for devlang**

The devlang allowlist contains valid language slugs for Markdown code blocks, such as `csharp` for C# code blocks. These slugs affect the labeling and colorization of the code blocks and new values must be approved by the Docs Information Architecture team. To request a new value, email [DevRella@service.microsoft.com](mailto:DevRella@service.microsoft.com).

## Request new attributes

If you need to track a new piece of metadata, you can request that we add it to our model. Please email [docsmetarequests@microsoft.com](mailto:docsmetarequests@microsoft.com) with a brief summary of the change you would like to make.

### TIP

Request a new attribute as soon as you think you will need it. It can often take two months (or more) to fully implement a new attribute in the reporting systems, because it requires allocating engineering resources.

## Other changes

To request other kinds of changes, email [docsmetarequests@microsoft.com](mailto:docsmetarequests@microsoft.com) with a brief summary of the change you would like to make.

## Metadata change process

You can see all requested changes that are currently in progress on the [metadata requests board](#).

Adding new values to the allowlists or changing fields for those values is a relatively quick and easy process and new values submitted by the end of day on Wednesday will usually be live by the following Friday. Requests for new attributes or taxonomies need to be more thoroughly vetted, and will take at least six weeks before they are implemented. Depending on the complexity of the request, it may be longer, but most reasonable requests can be accommodated, and we will work with you to balance our capabilities with the business needs behind the request.

For more information about metadata on Docs, contact [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

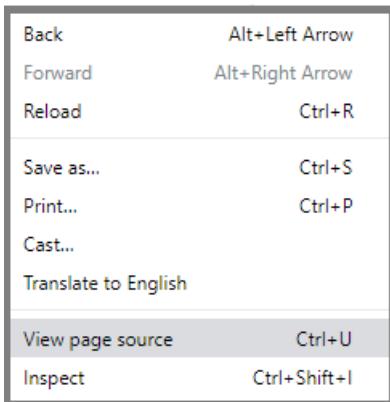
# Find the author of an article via metadata

4/26/2021 • 2 minutes to read

Internal Docs users at Microsoft sometimes want to find the author of an article to ask about the subject matter without suggesting edits to the article. Because author metadata can be set in multiple ways, the most reliable way to find the current author is to check the metadata on the published page.

## To check author metadata using HTML source code

1. Browse to the docs article in your web browser. In the page body, right-click to open a context menu. Select **View page source** or similar (exact wording differs by browser).



2. In the source view, find the `<meta name="author">` and `<meta name="ms.author">` HTML elements.

```
54 <meta name="ms.collection" content="collection" />
55 <meta name="author" content="rayne-wiselman" />
56 <meta name="ms.author" content="raynew" />
57 <meta name="ms.date" content="10/07/2019" />
```

3. The value of the `content` attribute in `<meta name="author">` is the author's GitHub ID. The value of the `content` attribute in `<meta name="ms.author">` is the author's Microsoft alias. Use the appropriate value to contact the author via the GitHub UI or via internal email.

For more information about how metadata is set, see [How to apply metadata](#).

## To check author metadata using Microsoft Docs Metrics (MDM)

1. Install the MDM browser extension by following [Install MDM into Microsoft Edge \(Chromium\) or Chrome](#).
2. Browse to the docs article in your web browser. Activate the MDM extension once you are on the page. You may be prompted to authenticate.
3. Once the MDM data renders, select **More Metadata** and on page that opens look for the **Author** tag. That corresponds to the Microsoft alias of the author.

# Metadata lists

3/5/2021 • 2 minutes to read

There are a few key central lists of metadata values that help us ensure high-quality metadata and accurate reporting on the docs.microsoft.com platform.

## Docs validation allowlists

See [Taxonomies for Docs](#).

The attributes are listed: `ms.topic`, `ms.devlang`, `ms.author`, `ms.service` + `ms.subservice`, `ms.prod` + `ms.technology`, `product`, etc. The brief view for each allowlist shows basic information such as the slug and associated label. For more information, click the [Detail View](#) link for each allowlist.

This allowlist is useful for you to look up certain required metadata attributes. When you use values outside of the allowed values in your docs, metadata validation warnings can appear at build time. For more information, see [Docs Build validation - Metadata](#).

This centralized list is kept in a database that helps back the metadata validation service enforce metadata across Docs.

## Business approver lists

There are text-based lists of the manager and business approvers in the contributor guide. These lists give PR Reviewers and other contributors a quick way to locate contact for escalations and issues that arise during docs authoring and publishing.

- [List of business approvers and repositories](#)
- [Business approver list for Azure services](#)

## How to update these lists

Content team managers are responsible for keeping the lists up to date. There are two places needing updates:

1. Follow the metadata changes steps to submit a request to update the metadata tracking database. See [How to request metadata changes](#).
2. Update the business approver list linked above by editing and issuing a PR to the contributor guide (docs-help-pr repo). Keep the table sorted alphabetically, ignoring the prefix words such as *Azure* in the sort.

# Taxonomies for Docs

5/14/2021 • 80 minutes to read

## Azure Category

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL
ai-machine-learning	AI + Machine Learning
analytics	Analytics
azure-virtual-desktop	Azure Virtual Desktop
blockchain	Blockchain
compute	Compute
containers	Containers
databases	Databases
developer-tools	Developer Tools
devops	DevOps
hybrid	Hybrid
identity	Identity
integration	Integration
iot	Internet of Things (IoT)
management-and-governance	Management and Governance
media	Media
migration	Migration
mixed-reality	Mixed Reality
mobile	Mobile
networking	Networking
security	Security

SLUG	LABEL
storage	Storage
web	Web
windows-virtual-desktop	Windows Virtual Desktop

## Certification Type

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL	DEFINITION
fundamentals	Fundamentals	Certification type class for foundational knowledge of Microsoft products (such as Microsoft Power Platform).
mce	MCE	Certification type class for Microsoft Certified Educator.
mcsa	MCSA	Certification type class for Microsoft Certified Solutions Associate.
mcsd	MCSD	Certification type class for Microsoft Certified Solutions Developer.
mcse	MCSE	Certification type class for Microsoft Certified Solutions Expert.
mos	MOS	Certification type class for Microsoft Office Specialist.
mta	MTA	Certification type class for Microsoft Technology Associate.
role-based	Role-based	Certification type class for roles associated with a Microsoft product (such as Azure Administrator).
specialty	Specialty	Certification type class for solution areas (such as Data & AI).

## Dev Lang

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL
al	AL
al-language	AL Language
aspx	ASPNET

SLUG	LABEL
aspx-csharp	ASP.NET (C#)
aspx-vb	ASP.NET (VB)
azcopy	AzCopy
azdata	Azure Data CLI
azsphere	Azure Sphere CLI
azurecli	Azure CLI
azurepowershell	Azure PowerShell
bash	Bash
bicep	Bicep
brainscript	BrainScript
c	C
console	Console
cpp	C++
cppcx	C++/CX
cppwinrt	C++/WinRT
csharp	C#
cshtml	CSHTML
d	D
dax	DAX
dockerfile	Dockerfile
dotnetcli	.NET CLI
fsharp	F#
go	Go
gradle	Gradle
haskell	Haskell

SLUG	LABEL
hcl	HashiCorp Configuration Language
hiveql	HiveQL
html	HTML
http	HTTP
inf	INF
inkling	Inkling
java	Java
javascript	JavaScript
js	JavaScript
json	JSON
kotlin	Kotlin
kusto	Kusto
lg	Bot response
lu	Language understanding
md	Markdown
mof	Managed Object Format
nodejs	Node.js
objc	Objective-C
odata	OData
output	Output
php	PHP
powerapps-comma	Power Apps
powerapps-dot	Power Apps
powerappsf1	PowerApps Formula
powershell	PowerShell

SLUG	LABEL
protobuf	ProtoBuf
python	Python
qna	Knowledge base
qsharp	Q#
r	R
ruby	Ruby
rust	Rust
scala	Scala
solidity	Solidity
sql	SQL
swift	Swift
terraform	Terraform
tsql	Transact-SQL
typescript	TypeScript
usql	U-SQL
vb	VB
vba	Visual Basic for Applications
vbscript	Visual Basic Script
xaml	XAML
xml	XML
xpp	X++
yaml	YAML

## Event Group

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL
azure-iaas-week	Azure IaaS Week
ces	CES
codemash	CodeMash
cpp-beyond	C++ and Beyond
devdays	DevDays
ghc	Grace Hopper Conference (GHC)
icse	International Conference on Software Engineering (ICSE)
ignite	Microsoft Ignite
microsoft-build	Microsoft Build
microsoft-firestarter	Microsoft FireStarter
microsoft-imagine-cup	Microsoft Imagine Cup
microsoft-lang-dot-net	Microsoft Lang.Net
microsoft-lang-dot-next	Microsoft Lang.Next
microsoft-mix	Microsoft MIX
microsoft-pdc	Microsoft Professional Developers Conference (PDC)
microsoft-spc	Microsoft SharePoint Connections (SPC)
microsoft-teched	Microsoft TechEd
microsoft-techfest	Microsoft TechFest
mms	Microsoft Management Summit (MMS)
splash	SPLASH
yow	YOW!

## Level

For more details, see the full taxonomy [Detail View](#).

SLUG	DEFINITION
advanced	Material that covers topics in depth as well as coverage of specialized topics.

SLUG	DEFINITION
beginner	Introductory and overview material that assumes little or no expertise with topic and covers topic concepts, functions, features, benefits.
intermediate	Material that assumes some knowledge but little in-depth understanding of the topic. Provides a detailed overview of a topic's sub-areas.

## Product

For more details, see the full taxonomy [Detail View](#).

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
altspacevr	AltspaceVR		
azure	Azure	azure-active-directory	Azure Active Directory
		azure-active-directory-b2c	Active Directory External Identities
		azure-active-directory-domain	Active Directory Domain Services
		azure-advisor	Azure Advisor
		azure-analysis-services	Analysis Services
		azure-anomaly-detector	Anomaly Detector
		azure-api-apps	API Apps
		azure-api-fhir	API for FHIR
		azure-api-management	API Management
		azure-app-configuration	App Configuration
		azure-application-gateway	Application Gateway
		azure-application-insights	Application Insights
		azure-app-service	App Service
		azure-app-service-mobile	App Service - Mobile Apps
		azure-app-service-static	App Service Static Web Apps
		azure-app-service-web	App Service - Web Apps

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-arc	Arc
		azure-archive-storage	Archive Storage
		azure-artifacts	Artifacts
		azure-attestation	Azure Attestation
		azure-automation	Automation
		azure-avere-vFXT	Avere vFXT
		azure-backup	Backup
		azure-bastion	Bastion
		azure-batch	Batch
		azure-bing-autosuggest	Bing Autosuggest
		azure-bing-custom	Bing Custom Search
		azure-bing-entity	Bing Entity Search
		azure-bing-image	Bing Image Search
		azure-bing-news	Bing News Search
		azure-bing-spellcheck	Bing Spell Check
		azure-bing-video	Bing Video Search
		azure-bing-visual	Bing Visual Search
		azure-bing-web	Bing Web Search
		azure-blob-storage	Blob Storage
		azure-blockchain-service	Blockchain Service
		azure-blockchain-tokens	Blockchain tokens
		azure-blockchain-workbench	Blockchain Workbench
		azure-blueprints	Blueprints
		azure-boards	Boards
		azure-bot-service	Bot Service

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-cache-redis	Cache for Redis
		azure-cdn	Content Delivery Network
		azure-clis	CLIs
		azure-cloud-services	Cloud Services
		azure-cloud-shell	Cloud Shell
		azure-cognitive-search	Cognitive Search
		azure-cognitive-services	Cognitive Services
		azure-communication-services	Communication Services
		azure-computer-vision	Computer Vision
		azure-container-instances	Container Instances
		azure-container-registry	Container Registry
		azure-content-moderator	Content Moderator
		azure-content-protection	Content Protection
		azure-cosmos-db	Cosmos DB
		azure-cost-management	Cost Management
		azure-custom-vision	Custom Vision
		azure-cyclecloud	CycleCloud
		azure-database-mariadb	Database for MariaDB
		azure-database-migration	Database Migration service
		azure-database-mysql	Database for MySQL
		azure-database-postgresql	Database for PostgreSQL
		azure-data-box-family	Data Box Family
		azure-databricks	Databricks
		azure-data-catalog	Data Catalog
		azure-data-explorer	Data Explorer

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-data-factory	Data Factory
		azure-data-lake	Data Lake
		azure-data-lake-analytics	Data Lake Analytics
		azure-data-lake-gen1	Data Lake Storage Gen1
		azure-data-lake-gen2	Data Lake Storage Gen2
		azure-data-lake-storage	Data Lake Storage
		azure-data-science-vm	Data Science Virtual Machines
		azure-data-share	Data Share
		azure-ddos-protection	DDos Protection
		azure-dedicated-host	Dedicated Host
		azure-dedicated-hsm	Dedicated HSM
		azure-devops	Azure DevOps
		azure-devops-tool-integrations	DevOps tool integrations
		azure-dev-spaces	Azure Dev Spaces
		azure-devtest-labs	DevTest Labs
		azure-dev-tool-integrations	Developer tool integrations
		azure-digital-twins	Digital Twins
		azure-disk-encryption	Disk Encryption
		azure-disk-storage	Disk Storage
		azure-dns	DNS
		azure-encoding	Encoding
		azure-event-grid	Event Grid
		azure-event-hubs	Event Hubs
		azure-expressroute	ExpressRoute

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-face	Face
		azure-farmbeats	FarmBeats
		azure-files	Files
		azure-firewall	Firewall
		azure-firewall-manager	Firewall Manager
		azure-form-recognizer	Form Recognizer
		azure-front-door	Front Door
		azure-functions	Functions
		azure-fxt-edge-filer	FXT Edge Filer
		azure-genomics	Microsoft Genomics
		azure-hdinsight	HDInsight
		azure-hdinsight-rserver	R Server for HDInsight
		azure-hpc-cache	HPC Cache
		azure-immersive-reader	Immersive Reader
		azure-information-protection	Azure Information Protection
		azure-ink-recognizer	Ink Recognizer
		azure-internet-analyzer	Internet Analyzer
		azure-iot	IoT
		azure-iot-central	IoT Central
		azure-iot-dps	IoT Device Provisioning Service
		azure-iot-edge	IoT Edge
		azure-iot-hub	IoT Hub
		azure-iot-pnp	IoT Plug and Play
		azure-iot-sdk	IoT SDK

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-iot-security-center	Security Center for IoT
		azure-iot-solution-accelerators	IoT Solution Accelerators
		azure-key-vault	Key Vault
		azure-kinect-dk	Kinect DK
		azure-kubernetes-service	Kubernetes Service
		azure-lab-services	Lab Services
		azure-language-understanding	Language Understanding
		azure-lighthouse	Lighthouse
		azure-linux-vm	Linux Virtual Machines
		azure-live-on-demand-streaming	Live and On-Demand Streaming
		azure-live-video-analytics	Live Video Analytics
		azure-load-balancer	Load Balancer
		azure-log-analytics	Log Analytics
		azure-logic-apps	Logic Apps
		azure-machine-learning	Machine Learning
		azure-machine-learning-designer	Machine Learning designer
		azure-machine-learning-studio	Machine Learning Studio
		azure-managed-applications	Managed Applications
		azure-managed-disks	Managed Disks
		azure-maps	Maps
		azure-media-analytics	Media Analytics
		azure-media-player	Media Player
		azure-media-services	Media Services

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-metrics-advisor	Metrics Advisor
		azure-migrate	Migrate
		azure-monitor	Monitor
		azure-netapp-files	Azure NetApp Files
		azure-network-watcher	Network Watcher
		azure-notebooks	Azure Notebooks
		azure-notification-hubs	Notification Hubs
		azure-open-datasets	Open Datasets
		azure-personalizer	Personalizer
		azure-pipelines	Pipelines
		azure-playfab	Playfab
		azure-policy	Policy
		azure-portal	Azure Portal
		azure-private-link	Private Link
		azure-qio	QIO
		azure-qna-maker	QnA Maker
		azure-quantum	Quantum
		azure-queue-storage	Queue Storage
		azure-rbac	Azure Role-based access control
		azure-redhat-openshift	Red Hat OpenShift
		azure-remote-rendering	Remote Rendering
		azure-repos	Repos
		azure-resource-graph	Resource Graph
		azure-resource-manager	Azure Resource Manager
		azure-rtos	RTOS

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-sap	SAP HANA on Azure Large Instances
		azure-scheduler	Scheduler
		azure-sdks	SDKs
		azure-search	Search
		azure-security-center	Azure Security Center
		azure-sentinel	Sentinel
		azure-service-bus	Service Bus
		azure-service-fabric	Service Fabric
		azure-service-health	Service Health
		azure-signalr-service	SignalR Service
		azure-site-recovery	Site Recovery
		azure-sovereign-china	Azure China 21Vianet
		azure-sovereign-germany	Azure Germany
		azure-sovereign-us	Azure US Government
		azure-spatial-anchors	Spatial Anchors
		azure-speaker-recognition	Speaker Recognition
		azure-speech	Speech
		azure-speech-text	Speech to Text
		azure-speech-translation	Speech Translation
		azure-sphere	Sphere
		azure-spring-cloud	Spring Cloud
		azure-sql-database	SQL Database
		azure-sql-edge	SQL Edge
		azure-sql-managed-instance	SQL Managed Instance

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-sqlserver-stretchdb	SQL Server Stretch Database
		azure-sqlserver-vm	SQL Server on Virtual Machines
		azure-sql-virtual-machines	SQL Virtual Machines
		azure-stack	Azure Stack
		azure-stack-edge	Azure Stack Edge
		azure-stack-hci	Azure Stack HCI
		azure-stack-hub	Azure Stack Hub
		azure-storage	Storage
		azure-storage-accounts	Storage Accounts
		azure-storage-explorer	Storage Explorer
		azure-storsimple	StorSimple
		azure-stream-analytics	Stream Analytics
		azure-synapse-analytics	Synapse Analytics
		azure-table-storage	Table Storage
		azure-test-plans	Test Plans
		azure-text-analytics	Text Analytics
		azure-text-speech	Text to Speech
		azure-time-series-insights	Time Series Insights
		azure-traffic-manager	Traffic Manager
		azure-translator	Translator
		azure-translator-speech	Translator Speech
		azure-translator-text	Translator Text
		azure-video-indexer	Video Indexer
		azure-virtual-desktop	Azure Virtual Desktop

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		azure-virtual-machines	Virtual Machines
		azure-virtual-machines-windows	Windows Virtual Machines
		azure-virtual-network	Virtual Network
		azure-virtual-wan	Virtual WAN
		azure-vm-scalesets	Virtual Machine Scale Sets
		azure-vmware-solution	Azure VMware Solution
		azure-vpn-gateway	VPN Gateway
		azure-webapp-containers	Web App for Containers
		azure-web-application-firewall	Web Application Firewall
		azure-web-apps	Web Apps
		windows-azure-pack	Windows Azure Pack
bing	Bing		
blazor	Blazor	blazor-server	Blazor Server
		blazor-webassembly	Blazor WebAssembly
clarity	Clarity		
connected-services-framework	Microsoft Connected Services Framework		
consumer	Consumer		
customer-care-framework	Customer Care Framework		
dotnet	.NET	aspnet	ASP.NET
		aspnet-core	ASP.NET Core
		dotnet-core	.NET Core
		dotnet-standard	.NET Standard
		ef6	Entity Framework 6.0
		ef-core	Entity Framework Core

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		mlnet	ML.NET
		xamarin	Xamarin
dynamics	Dynamics		
dynamics-365	Dynamics 365	customer-voice	Customer Voice
		dynamics-365-import-tool	Dynamics 365 Import Tool
		dynamics-business-central	Business Central
		dynamics-commerce	Commerce
		dynamics-cust-insights	Customer Insights
		dynamics-customer-engagement	Customer Engagement apps
		dynamics-customer-service	Customer Service
		dynamics-cust-svc-insights	Customer Service Insights
		dynamics-field-service	Field Service
		dynamics-finance	Finance
		dynamics-finance-operations	Finance and Operations
		dynamics-fraud-protection	Fraud Protection
		dynamics-guides	Guides
		dynamics-human-resources	Human Resources
		dynamics-iom	Dynamics 365 Intelligent Order Management
		dynamics-layout	Layout
		dynamics-marketing	Marketing
		dynamics-market-insights	Market Insights
		dynamics-product-insights	Product Insights
		dynamics-prod-visualize	Product Visualize
		dynamics-project-operations	Project Operations

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		dynamics-project-service	Project Service Automation
		dynamics-remote-assist	Remote Assist
		dynamics-retail	Retail
		dynamics-sales	Sales
		dynamics-sales-insights	Sales Insights
		dynamics-scm	Supply Chain Management
		dynamics-talent	Talent
		dynamics-talent-attract	Talent Attract
		dynamics-talent-core	Talent Core
		dynamics-talent-onboard	Talent Onboard
expression	Expression	expression-studio	Expression Studio
flipgrid	Flipgrid		
github	GitHub		
hololens	HoloLens		
industry-solutions	Industry Solutions	mscloud-financial	Microsoft Cloud for Financial Services
		mscloud-healthcare	Microsoft Cloud for Healthcare
		mscloud-manufacturing	Microsoft Cloud for Manufacturing
		mscloud-nonprofit	Microsoft Cloud for Nonprofit
		mscloud-retail	Microsoft Cloud for Retail
		return-to-school	Return to School
		return-to-workplace	Return to the Workplace
internet-explorer	Internet Explorer		
kinect	Kinect		
m365	Microsoft 365	fluid-framework	Fluid Framework

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		m365-ems	Enterprise Mobility + Security
		m365-ems-cloud-app-security	Cloud App Security
		m365-information-protection	Microsoft Information Protection
		m365-myanalytics	MyAnalytics
		m365-security-center	Microsoft 365 Security Center
		m365-security-score	Security Score
		m365-threat-protection	Threat Protection
		m365-workplace-analytics	Workplace Analytics
makecode	Microsoft MakeCode		
mdatp	Microsoft Defender for Endpoint		
mem	Microsoft Endpoint Manager	mem-configuration-manager	Endpoint Configuration Manager
		mem-intune	Intune
microsoft-authentication-library	Microsoft Authentication Library	microsoft-identity-web	Microsoft Identity Web Authentication Library
		msal-android	Microsoft Authentication Library for Android
		msal-angular	Microsoft Authentication Library for Angular
		msal-ios	Microsoft Authentication Library for iOS
		msal-java	Microsoft Authentication Library for Java
		msal-js	Microsoft Authentication Library for JavaScript
		msal-node	Microsoft Authentication Library for Node
		msal-python	Microsoft Authentication Library for Python

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		passport-azure-ad	Microsoft Azure Active Directory Passport.js Plug-in
microsoft-edge	Microsoft Edge		
microsoft-mesh	Microsoft Mesh		
microsoft-servers	Microsoft Servers		
minecraft	Minecraft		
mrtk	Mixed Reality Toolkit		
ms-build-openjdk	Microsoft Build of OpenJDK		
msc	Microsoft System Center	m365-ems-configuration-manager	System Center Configuration Manager
		msc-operations-manager	Operations Manager
		msc-service-manager	Service Manager
ms-graph	Microsoft Graph		
office	Office	office-365-atp	Advanced Threat Protection
		office-access	Access
		office-adaptive-cards	Adaptive Cards
		office-add-ins	Office Add-ins
		office-bookings	Bookings
		office-excel	Excel
		office-exchange-server	Exchange Server
		office-forefront	Forefront
		office-kaizala	Kaizala
		office-lync-server	Lync Server
		office-onedrive	OneDrive
		office-onenote	OneNote
		office-outlook	Outlook

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		office-planner	Planner
		office-powerpoint	PowerPoint
		office-project	Project
		office-project-server	Project Server
		office-publisher	Publisher
		office-skype-business	Skype for Business
		office-sp	SharePoint
		office-sp-designer	SharePoint Designer
		office-sp-framework	SharePoint Framework
		office-sp-server	SharePoint Server
		office-ui-fabric	Office UI Fabric
		office-visio	Visio
		office-word	Word
		office-yammer	Yammer
		sway	Sway
office-365	Office 365		
office-teams	Microsoft Teams		
power-platform	Power Platform	ai-builder	AI Builder
		azure-powerbi-embedded	Power BI Embedded
		common-data-service	Microsoft Dataverse
		power-apps	Power Apps
		power-automate	Power Automate
		power-bi	Power BI
		power-query	Power Query
		power-virtual-agents	Power Virtual Agents

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
project-acoustics	Project Acoustics		
qdk	Quantum Development Kit		
silverlight	Silverlight		
skype	Skype		
sql-server	SQL Server	sql-server-2008	SQL Server 2008
surface	Surface	surface-duo	Surface Duo
vs	Visual Studio	vs-app-center	App Center
		vs-code	Visual Studio Code
		vs-mac	Visual Studio for Mac
		vs-online	Visual Studio Online
windows	Windows	windows-api-win32	Windows API - Win32
		windows-forms	Windows Forms
		windows-iot	Windows IoT
		windows-iot-10core	Windows 10 IoT Core
		windows-mdop	Desktop Optimization Pack
		windows-mixed-reality	Windows Mixed Reality
		windows-server	Windows Server
		windows-smb-server	Windows Small Business Server
		windows-system-center	Windows System Center
		windows-uwp	Universal Windows Platform (UWP)
		windows-virtual-desktop	Windows Virtual Desktop
		windows-wdk	Windows Driver Kit (WDK)
		windows-wpf	Windows Presentation Foundation (WPF)
xbox	Xbox		

# Role

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL	DEFINITION
administrator	Administrator	Individuals who provide proactive administrative support to a manager and team.
ai-edge-engineer	AI Edge Engineer	People who have the ability to deploy AI and ML models in containers at the Edge.
ai-engineer	AI Engineer	People who analyze requirements for cloud-based or hybrid AI needs and implement solutions.
auditor	Auditor	People who analyze and assess the adequacy and effectiveness of security controls implemented by a Cloud Service Provider's IT systems
business-analyst	Business Analyst	Individuals who gather, develop, review and/or prioritize requirements, projections and/or specifications (technical or otherwise) about an application, system, process, department, or organization.
business-owner	Business Owner	People who have the authority to make decisions about a business area or division within a company.
business-user	Business User	People who use computers or computer software in order to perform business transactions.
data-analyst	Data Analyst	A person who uses Power BI to help an organization analyze data, and create report and dashboards.
database-administrator	Database Administrator	The people responsible for managing databases.
data-engineer	Data Engineer	Individuals whose responsibility it is to gather, store, process, and serve data, so that data scientists can easily query it.
data-scientist	Data Scientist	Individuals who use scientific methods, processes, algorithms, and systems to extract knowledge and insights from structured and unstructured data in various forms.

SLUG	LABEL	DEFINITION
developer	Developer	Individuals who design, develop, test, and evaluate software and systems that make computers work.
devops-engineer	DevOps Engineer	People who work with developers, system operators, and other production IT staff to oversee code releases.
functional-consultant	Functional Consultant	People that analyze and translate business requirements into solutions that meet the requirements of a business.
higher-ed-educator	Higher Education Educator	Individuals that work in an instructional capacity at an institution of higher education (college or university).
identity-access-admin	Identity and Access Administrator	Individuals that specialize in driving strategic identity projects, modernizing identity solutions, implementing hybrid identity solutions, and identity governance.
ip-admin	Information Protection Administrator	Individuals that define applicable requirements and test IT processes and operations against those policies and controls.
k-12-educator	K-12 Educator	Individuals that work in an instructional capacity at a primary or secondary learning institution (elementary, middle, or high school).
maker	App Maker	A non-developer who uses PowerApps and Microsoft Flow to create apps to quickly improve business processes.
network-engineer	Network Engineer	A person who designs, implements, and supports local area and wide area networks within an organization.
parent-guardian	Parent/Guardian	An individual who is legally responsible for a person who is under the age of legal competence.
privacy-manager	Privacy Manager	People who develop, implement, and enforce policies and procedures as part of a Cloud Service Provider's privacy programs.
risk-practitioner	Risk Practitioner	People who identify and analyze risks associated with a Cloud Service Provider's IT systems.

SLUG	LABEL	DEFINITION
school-leader	School Leader	Individual who works as an administrator at an educational institution.
security-engineer	Security Engineer	Security Engineers implement security controls and threat protection, manage identity and access, and protect data, applications, and networks in cloud and hybrid environments as part of end-to-end infrastructure.
security-operations-analyst	Security Operations Analyst	Individuals that specialize in running and analyzing the security operations center of a company.
service-adoption-specialist	Service Adoption Specialist	Individuals who drive adoption of Microsoft services within their organization.
solution-architect	Solution Architect	Individuals whose main responsibility is to define the application architecture (what set of modules is needed for a project and through what protocols they will be interconnected) for developers in the enterprise.
student	Student	Someone who is studying to learn new technology skills. They may not yet have a specialization in mind that would fit them into another role, or they may be interested in topics that are relevant to multiple types of roles.
technology-manager	Technology Manager	People who are responsible for managing and directing a company's technology infrastructure.

## Salesplay

The `salesplay` taxonomy aligns to standard OCP "Solution Area" taxonomy. This taxonomy is updated annually at the start of the fiscal year. The values listed below are accurate for FY21.

For more details, see the full taxonomy [Detail View](#).

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
azure	Azure	advanced-networking	Advanced Networking
		analytics-ai	Analytics & AI
		azure-vmware-solutions	Azure VMWare Solutions
		blockchain	Blockchain

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		business-continuity-disaster-recovery	Business Continuity & Disaster Recovery
		cloud-native-apps	Cloud Native Apps with AI, Kubernetes, Azure Cosmos DB / PostgreSQL
		cloud-to-cloud-migration	Cloud to Cloud Migration
		devops-with-github	DevOps with GitHub
		enable-azure	Enable Azure
		gaming	Gaming
		hpc	HPC
		hybrid-cloud-solutions	Hybrid Cloud Solutions
		iot	IoT
		linux-oss-dbs	Linux & OSS DBs
		mainframe-migration	Mainframe Migration
		mixed-reality	Mixed Reality
		modernize-dotnet-apps	Modernize .NET Apps with App Service, Azure SQL DB
		sap-on-azure	SAP on Azure
		storage-file-systems	Storage & File Systems
		windows-server-sql-server	Windows Server & SQL Server
		windows-virtual-desktop	Windows Virtual Desktop (WVD)
business-applications	Business Applications	activate-digital-selling	Activate Digital Selling
		build-agile-business-processes	Build Agile Business Processes
		build-a-resilient-supply-chain	Build a Resilient Supply Chain
		business-management-for-smbs	Business Management for SMBs
		enable-always-on-service	Enable Always-On Service

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		generate-value-with-proactive-insights	Generate Value with Proactive Insights
		manage-financial-risk	Manage Financial Risk and Reduce Fraud
modern-work-security	Modern Work & Security	compliance	Compliance
		firstline-workers	Firstline Workers
		knowledge-insights	Knowledge & Insights
		secure-remote-work	Secure Remote Work
		security	Security
		surface-partner-devices	Surface & Partner Devices
		teams-chat-collab	Teams Chat & Collab
		teams-meetings-calling-devices	Teams Meetings, Calling & Devices
		teams-platform	Teams Platform
		workplace-analytics	Workplace Analytics

## ms.author

The `ms.author` must be a valid individual Microsoft alias or an approved distribution list (DL).

To request a new approved DL, email [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

For more details, see the full taxonomy [Detail View](#).

MS.AUTHOR DL
amlstudiodocs
apimpm
archiveddocs
azfuncdf
azurespheredocs
betafred
daxcpft

MS.AUTHOR\_DL

dotnetcontent

hisdocs

hvdev

msedgedevrel

msgraphdocsvtteam

ncldev

o365devx

office365servicedesc

pnp

tdsp

wcfsrvt

wdg-dev-content

windows-driver-content

windowsdriverdev

windowssdkdev

xamadodi

## ms.devlang

The `ms.devlang` allowlist contains valid language slugs for Markdown code blocks, such as `csharp` for C# code blocks. These slugs affect the labeling and colorization of the code blocks and new values must be approved by the Docs Information Architecture team.

To request a new value, email [DevRella@service.microsoft.com](mailto:DevRella@service.microsoft.com).

For more details, see the full taxonomy [Detail View](#).

SLUG	LABEL
azurecli	Azure CLI
azure-sphere-cli	Azure Sphere CLI
bicep	Bicep
brainscript	BrainScript

SLUG	LABEL
c	C
cli	CLI
cpp	C++
csharp	C#
dotnet	.NET
fsharp	F#
go	Go
java	Java
javascript	JavaScript
json	JSON
lg	Bot response
lu	Language understanding
nodejs	Node.js
objective-c	Objective-C
php	PHP
powershell	PowerShell
python	Python
qna	Knowledge base
qsharp	Q#
r	R
rest-api	REST API
ruby	Ruby
rust	Rust
spark-scala	Spark Scala
swift	Swift

SLUG	LABEL
typescript	TypeScript
vb	Visual Basic
vstscli	VSTS CLI

## ms.prod

The `ms.prod` value indicates the product an article applies to; `ms.technology` provides more granular detail about the specified product and can only be used if `ms.prod` is also used. The following table shows valid pairs of `ms.prod` and `ms.technology`. An `ms.technology` value of "(empty)" means `ms.prod` can be used as standalone metadata without a child `ms.technology`.

To request a new single `ms.prod / ms.technology` pair, fill out [the form for new ms.prod/technology values](#). Requests submitted by end-of-day Thursday will be processed on Friday and live in all systems by Monday.

For more details, see the full taxonomy [Detail View](#).

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
access	Access	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		connectors	Connectors
		custom-web-app	Custom web app
		data-visualizations	Data visualizations
		desktop-database	Desktop database
		fast-search	FAST search
		ms-bot-framework	Microsoft Bot Framework
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		tabs	Tabs
		vba	VBA
adaptive-cards	Adaptive Cards	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
advanced-threat-analytics	Advanced Threat Analytics	(empty)	(empty)
analysis-services	SQL Analysis Services	(empty)	(empty)
		mpp-data-warehouse	MPP Data Warehouse
analytics	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
analytics-platform-system	Analytics Platform System	(empty)	(empty)
		mpp-data-warehouse	MPP Data Warehouse
applications	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
asp.net	ASPNET	(empty)	(empty)
asp.net-4.0	ASPNET 4.0	(empty)	(empty)
asp.net-4.5.1	ASPNET 4.5.1	(empty)	(empty)
asp.net-5.0	ASPNET 5.0	(empty)	(empty)
asp.net-mvc-3.0	ASPNET MVC 3.0	(empty)	(empty)
asp.net-mvc-5.0	ASPNET MVC 5.0	(empty)	(empty)
		aspnetcore	ASPNET Core index page
		aspnetcore-azure	ASPNET Core with Azure
		aspnetcore-blazor	ASPNET Core Blazor
		aspnetcore-clientside	ASP.NET Core client-side
		aspnetcore-data	ASPNET Core data access
		aspnetcore-fundamentals	ASPNET Core fundamentals
		aspnetcore-getstarted	ASPNET Core get started
		aspnetcore-grpc	ASPNET Core gRPC
		aspnetcore-hostdeploy	ASPNET Core hosting and deployment
		aspnetcore-migration	ASPNET Core migration

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		aspnetcore-mobile	ASP.NET Core mobile
		aspnetcore-mvc	ASP.NET Core MVC
		aspnetcore-performance	ASP.NET Core performance
		aspnetcore-razorpages	ASP.NET Core Razor Pages
		aspnetcore-releasenotes	ASP.NET Core release notes
		aspnetcore-security	ASP.NET Core security
		aspnetcore-signalr	ASP.NET Core SignalR
		aspnetcore-test	ASP.NET Core testing
		aspnetcore-tutorials	ASP.NET Core tutorials
		aspnetcore-webapi	ASP.NET Core Web API
		aspnetcore-whatsnew	ASP.NET Core What's new pages
aspnet-core	ASP.NET Core	(empty)	(empty)
		aspnetcore	ASP.NET Core index page
		aspnetcore-azure	ASP.NET Core with Azure
		aspnetcore-blazor	ASP.NET Core Blazor
		aspnetcore-clientside	ASP.NET Core client-side
		aspnetcore-data	ASP.NET Core data access
		aspnetcore-fundamentals	ASP.NET Core fundamentals
		aspnetcore-getstarted	ASP.NET Core get started
		aspnetcore-grpc	ASP.NET Core gRPC
		aspnetcore-hostdeploy	ASP.NET Core hosting and deployment
		aspnetcore-migration	ASP.NET Core migration
		aspnetcore-mobile	ASP.NET Core mobile
		aspnetcore-mvc	ASP.NET Core MVC

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		aspnetcore-performance	ASP.NET Core performance
		aspnetcore-razorpages	ASP.NET Core Razor Pages
		aspnetcore-releasenotes	ASP.NET Core release notes
		aspnetcore-security	ASP.NET Core security
		aspnetcore-signalr	ASP.NET Core SignalR
		aspnetcore-test	ASP.NET Core testing
		aspnetcore-tutorials	ASP.NET Core tutorials
		aspnetcore-webapi	ASP.NET Core Web API
		aspnetcore-whatsnew	ASP.NET Core What's new pages
aspnet-core-api	ASP.NET Core API Reference	(empty)	(empty)
aspnet-framework	ASP.NET Framework	(empty)	(empty)
		aspnet-ajax	ASP.NET 4.x AJAX
		aspnet-aspNet	ASP.NET 4.x overview
		aspnet-identity	ASP.NET 4.x Identity
		aspnet-mobile	ASP.NET 4.x mobile
		aspnet-mvc	ASP.NET 4.x MVC
		aspnet-signalr	ASP.NET 4.x SignalR
		aspnet-spa	ASP.NET 4.x SPA
		aspnet-visualstudio	ASP.NET 4.x Visual Studio
		aspnet-webapi	ASP.NET 4.x Web API
		aspnet-webforms	ASP.NET 4.x Web Forms
		aspnet-webhooks	ASP.NET 4.x Web Hooks
		aspnet-webpages	ASP.NET 4.x Web Pages
		aspnet-whitepapers	ASP.NET 4.x Whitepapers
aspnet-mvc-api	ASP.NET MVC API Reference	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
aspnet-webpages-api	ASP.NET Web Pages API Reference	(empty)	(empty)
azure	Azure	(empty)	(empty)
		azure-cli	Azure CLI
		azure-data-studio	Azure Data Studio
		azure-sdk-go	Azure SDK for Go
		machine-learning	Machine Learning
azure-data-studio	Azure Data Studio	(empty)	(empty)
		azure-data-studio	Azure Data Studio
		machine-learning	Machine Learning
azure-gaming	Microsoft Azure Gaming	(empty)	(empty)
azure-java	Java	(empty)	(empty)
azure-nodejs	JavaScript	(empty)	(empty)
azure-private-endpoint	Private Endpoint	(empty)	(empty)
azure-private-link-service	Private Link service	(empty)	(empty)
azure-python	Python	(empty)	(empty)
azure-sphere	Azure Sphere	(empty)	(empty)
		bing-ads-api	Bing Ads API
		bing-ads-hotel-ads	Hotel Ads
		bing-ads-scripts	Bing Ads Scripts
		bing-ads-sdk	Bing Ads SDK
		bing-ads-shopping-content	Content API
bing-ads	Bing Ads	(empty)	(empty)
		bing-ads-api	Bing Ads API
		bing-ads-hotel-ads	Hotel Ads
		bing-ads-scripts	Bing Ads Scripts

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		bing-ads-sdk	Bing Ads SDK
		bing-ads-shopping-content	Content API
biztalk-server	BizTalk Server	(empty)	(empty)
biztalk-server-2013	BizTalk Server 2013	(empty)	(empty)
		microsoft-graph	Microsoft Graph
bookings	Bookings	(empty)	(empty)
		microsoft-graph	Microsoft Graph
bot-framework	Bot Framework	(empty)	(empty)
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		content-pack	Content pack
		data-visualizations	Data visualizations
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		tabs	Tabs
business-platform	Business platform	(empty)	(empty)
		content-pack	Content pack
		data-visualizations	Data visualizations
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		tabs	Tabs
calendar	Outlook	(empty)	(empty)
		microsoft-graph	Microsoft Graph
call	Non Product Specific	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		microsoft-graph	Microsoft Graph
call-records	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
change-notifications	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
chaosstudio	Azure Chaos Studio	(empty)	(empty)
		chaosstudio	Azure Chaos Studio
cloud-communications	Office	(empty)	(empty)
		microsoft-graph	Microsoft Graph
cloud-pc	Cloud PC API	(empty)	(empty)
		microsoft-graph	Microsoft Graph
cloud-printing	Universal Print	(empty)	(empty)
		microsoft-graph	Microsoft Graph
cntk	Cognitive Toolkit	(empty)	(empty)
collaborate	MSCollaborate	(empty)	(empty)
communications-server	Communications Server	(empty)	(empty)
communications-server-2003	Communications Server 2003	(empty)	(empty)
communications-server-2005	Communications Server 2005	(empty)	(empty)
communications-server-2007	Communications Server 2007	(empty)	(empty)
communications-server-2007-r2	Communications Server 2007 R2	(empty)	(empty)
communicator	Communicator	(empty)	(empty)
community-toolkit	Windows Community Toolkit	(empty)	(empty)
compliance	Non Product Specific	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		microsoft-graph	Microsoft Graph
configuration-manager	Configuration Manager	(empty)	(empty)
		configmgr-analytics	Desktop Analytics
		configmgr-app	App Management
		configmgr-client	Client Management
		configmgr-comanage	Co-management
		configmgr-compliance	Compliance
		configmgr-core	Core infrastructure
		configmgr-hybrid	Hybrid
		configmgr-mdm	Mobile device management
		configmgr-mdt	Microsoft Deployment Toolkit
		configmgr-osd	Operating System Deployment
		configmgr-other	Other
		configmgr-protect	Protect
		configmgr-sdk	SDK
		configmgr-sum	Software Updates
configuration-manager-2012	Configuration Manager 2012	(empty)	(empty)
configuration-manager-2012-r2	Configuration Manager 2012	(empty)	(empty)
configuration-manager-2012-sp1	Configuration Manager 2012	(empty)	(empty)
connected-devices	Connected Devices	(empty)	(empty)
console	Console	(empty)	(empty)
corporate-management	Intune	(empty)	(empty)
		microsoft-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
cortana	Cortana	(empty)	(empty)
		devlang-csharp	Azure attach
		skills	Cortana skills
cpp-dev15	Visual C++ 2017	(empty)	(empty)
		devlang-csharp	Azure attach
crm-2011	Dynamics CRM 2011	(empty)	(empty)
crm-2013	Dynamics CRM 2013	(empty)	(empty)
crm-2015	Dynamics CRM 2015	(empty)	(empty)
crm-2016	Dynamics CRM 2016	(empty)	(empty)
cross-device-experiences	Project Rome	(empty)	(empty)
		microsoft-graph	Microsoft Graph
customer-booking	Bookings	(empty)	(empty)
		microsoft-graph	Microsoft Graph
cyclecloud	Azure CycleCloud	(empty)	(empty)
		microsoft-graph	Microsoft Graph
data-access	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
data-connect	Microsoft Graph data connect	(empty)	(empty)
		microsoft-graph	Microsoft Graph
deep-zoom-composer	z_Deep Zoom Composer	(empty)	(empty)
delve	Delve	(empty)	(empty)
deployment-manager	Microsoft Deployment Manager	(empty)	(empty)
desktop	Win32 and other desktop technologies	(empty)	(empty)
		accessibility	Accessibility
		audio-video	Audio and video

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		compatibility-cookbook	Compatibility cookbook
		data-access-storage	Data access and storage
		deployment	Deployment
		design	Design
		desktop-app-ui	Desktop app ui
		desktop-environment	Desktop environment
		devices	Devices
		diagnostics	Diagnostics
		documents-printing	Documents and printing
		get-started	Get started
		graphics-gaming	Graphics and gaming
		installation-servicing	Application installation and servicing
		machine-learning	Machine learning
		networking	Networking and internet
		sdk-api-reference	Windows SDK
		security-identity	Security and identity
		server-tech	Server technologies
		system-services	System services
dev-environment	Dev Environment	(empty)	(empty)
		package-manager	Windows Package Manager
		powertoys	PowerToys
		windows-android	Windows Android
		windows-javascript	Windows JavaScript
		windows-nodejs	Windows NodeJS
		windows-python	Windows Python

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows-rust	Windows Rust
		windows-subsystem-for-linux	Windows Subsystem for Linux
		windows-terminal	Windows Terminal
devices-and-apps	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
devops	Azure DevOps	(empty)	(empty)
		devops-accounts	Azure DevOps Accounts
		devops-agile	Azure Boards
		devops-analytics	Azure DevOps Analytics
		devops-artifacts	Azure Artifacts
		devops-audit	Azure DevOps Audit
		devops-billing	Azure DevOps Lbilling
		devops-cicd	Azure Pipelines
		devops-cicd-actions	Pipeline actions
		devops-cicd-agents	Pipeline agents
		devops-cicd-apps	Pipeline apps
		devops-cicd-archive	Pipeline archive
		devops-cicd-artifacts	Pipeline artifacts
		devops-cicd-build	Pipeline build
		devops-cicd-caching	Pipeline caching
		devops-cicd-ecosystems	Pipeline ecosystems
		devops-cicd-get-started	Pipeline started
		devops-cicd-integrations	Pipeline integrations
		devops-cicd-library	Pipeline library
		devops-cicd-licensing	Pipeline licensing

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		devops-cicd-media	Pipeline media
		devops-cicd-migrate	Pipeline migrate
		devops-cicd-packages	Pipeline packages
		devops-cicd-policies	Pipeline policies
		devops-cicd-process	Pipeline process
		devops-cicd-release	Pipeline release
		devops-cicd-reports	Pipeline reports
		devops-cicd-repos	Pipeline repos
		devops-cicd-scripts	Pipeline scripts
		devops-cicd-security	Pipeline security
		devops-cicd-targets	Pipeline targets
		devops-cicd-tasks	Pipeline tasks
		devops-cicd-test	Pipeline test
		devops-code-git	Azure Repos (Git)
		devops-code-tfvc	Azure Repos (TFVC)
		devops-collab	Azure DevOps Collaboration
		devops-cross-platform	Azure DevOps Cross Platform
		devops-ecosystem	Azure DevOps Developer Docs
		devops-learn	DevOps Resource Center
		devops-marketplace	Azure DevOps Marketplace
		devops-migrate	Azure DevOps Migration
		devops-new-user	Azure DevOps New User Guide
		devops-projects	Azure DevOps Organizations

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		devops-public-projects	Azure DevOps Public Projects
		devops-ref	Azure DevOps API Reference
		devops-reference	Azure DevOps Reference
		devops-release-notes	Azure DevOps Release Notes
		devops-security	Azure DevOps Security
		devops-settings	Azure DevOps Settings
		devops-test	Azure Test Plans
		devops-whitepapers	Azure DevOps Articles
		vs-devops-package	Visual Studio DevOps Package
devops-server	TFs and Codex Server	(empty)	(empty)
		tfs-admin	Codex Server Administration
		tfs-migration-guide	Codex Server/TFS Migration Guide
directory-management	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
dotnet-api	.NET API Reference	(empty)	(empty)
		accessibility	Accessibility Namespace
		dbling.util	DbLinq.Util Namespace
		microsoft.activities	Microsoft.Activities Namespaces
		microsoft.build	Microsoft.Build Namespaces
		microsoft.csharp	Microsoft.CSharp Namespaces
		microsoft.jscript	Microsoft.JScript Namespaces

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		microsoft.sqlserver	Microsoft.SqlServer Namespaces
		microsoft.visualbasic	Microsoft.VisualBasic Namespaces
		microsoft.visualc	Microsoft.VisualC Namespaces
		microsoft.win32	Microsoft.Win32 Namespaces
		microsoft.windows	Microsoft.Windows Namespaces
		mono	Mono Namespaces
		system	System Namespace
		system.activities	System.Activities Namespaces
		system.addin	System.AddIn Namespaces
		system.buffers	System.Buffers Namespaces
		system.codedom	System.CodeDom Namespaces
		system.collections	System.Collections Namespaces
		system.componentmodel	System.ComponentModel Namespaces
		system.composition	System.Composition Namespaces
		system.configuration	System.Configuration Namespaces
		system.data	System.Data Namespaces
		system.deployment	System.Deployment Namespaces
		system.device	System.Device Namespaces
		system.diagnostics	System.Diagnostics Namespaces
		system.directoryservices	System.DirectoryServices Namespaces

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		system.drawing	System.Drawing Namespaces
		system.dynamic	System.Dynamic Namespaces
		system.enterpriseservices	System.EnterpriseServices Namespaces
		system.globalization	System.Globalization Namespaces
		system.identitymodel	System.IdentityModel Namespaces
		system.io	System.IO Namespaces
		system.json	System.Json Namespaces
		system.linq	System.Linq Namespaces
		system.management	System.Management Namespaces
		system.media	System.Media Namespaces
		system.messaging	System.Messaging Namespaces
		system.net	System.Net Namespaces
		system.numerics	System.Numerics Namespaces
		system.printing	System.Printing Namespaces
		system.reflection	System.Reflection Namespaces
		system.resources	System.Resources Namespaces
		system.runtime	System.Runtime Namespaces
		system.security	System.Security Namespaces
		system.servicemodel	System.ServiceModel Namespaces

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		system.serviceprocess	System.ServiceProcess Namespaces
		system.speech	System.Speech Namespaces
		system.text	System.Text Namespaces
		system.threading	System.Threading Namespaces
		system.timers	System.Timers Namespaces
		system.transactions	System.Transactions Namespaces
		system.web	System.Web Namespaces
		system.windows	System.Windows Namespaces
		system.workflow	System.Workflow Namespaces
		system.xaml	System.Xaml Namespaces
		system.xml	System.Xml Namespaces
		uiautomationclientsideproviders	UIAutomationClientsideProviders Namespace
		xamlgeneratednamespace	XamlGeneratedNamespace Namespace
dotnet-architecture	.NET Architecture e-books	(empty)	(empty)
		blazor	Blazor for Web Forms developers
		cloud-native	Cloud Native
		containerized-lifecycle	.NET Microservices - DevOps
		dapr	.NET DAPR applications
		grpc	gRPC for WCF developers
		microservices	.NET microservices - Architecture
		modernize-desktop-apps	Modernizing desktop apps

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		modernize-with-azure-containers	Modernizing .NET apps
		modern-web-apps-azure	Modern ASP.NET Web applications
		porting-asp-to-core	Port ASP.NET to ASP.NET Core
		serverless	Serverless apps
dotnet-azure	.NET Azure development	(empty)	(empty)
dotnet-core	.NET Core	(empty)	(empty)
dotnet-csharp	.NET - C#	(empty)	(empty)
		csharp-advanced-concepts	C# advanced concepts
		csharp-async	Async task programming
		csharp-diagnostics	Errors and Warnings
		csharp-fundamentals	C# fundamentals
		csharp-get-started	C# get started
		csharp-language-reference	C# language reference
		csharp-linq	C# LINQ
		csharp-null-safety	C# nullable reference types
		csharp-roslyn	Roslyn SDK
dotnet-data	.NET Data Access	csharp-spec	C# Language Spec
		csharp-whats-new	C# what's new
		(empty)	(empty)
		dotnet-ado	.NET ADO
dotnet-data-prep	Microsoft Data Prep .NET SDK	dotnet-ef	.NET Entity Framework
		dotnet-sql-client	.NET SQL Client
dotnet-desktop	.NET Desktop	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dotnet-winforms	Winforms
		dotnet-wpf	WPF
dotnet-framework	.NET Framework	(empty)	(empty)
		dotnet-appcompat	App Compat
		dotnet-data	Data Access
		dotnet-docker	Docker
		dotnet-install	Installation and deployment
		dotnet-networking	Networking
		dotnet-security	Security
		dotnet-wcf	WCF
		dotnet-wf	Windows Workflow
		dotnet-winforms	Winforms
		dotnet-wpf	WPF
dotnet-fsharp	.NET - F#	(empty)	(empty)
dotnet-fundamentals	.NET - Fundamentals	(empty)	(empty)
		dotnet-analyzers	.NET code analyzers
		dotnet-cli	.NET command line interface
		dotnet-docker	.NET Docker containers
		dotnet-reactive-extensions	.NET reactive extensions
dotnet-gaming	.NET Game development	(empty)	(empty)
dotnet-iot	.NET IoT development	(empty)	(empty)
dotnet-ml	.NET - ML.NET	(empty)	(empty)
dotnet-ml-api	.NET - ML.NET API Reference	(empty)	(empty)
dotnet-mobile	.NET Mobile development	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dotnet-android	.NET for Android
		dotnet-essentials	.NET MAUI Essentials
		dotnet-ios	.NET for iOS
		dotnet-mac	.NET for macOS
		dotnet-maui	.NET MAUI
		dotnet-skiasharp	.NET SkiaSharp
dotnet-roslyn-api	.NET - Roslyn SDK API Reference	(empty)	(empty)
		microsoft.codeanalysis	Microsoft.CodeAnalysis namespaces
dotnet-spark	.NET for Apache SPARK	(empty)	(empty)
dotnet-visualbasic	.NET - Visual Basic	(empty)	(empty)
		vb-diagnostics	Errors and Warnings
		vb-spec	VB Language Spec
dotnet-web	.NET Web development	(empty)	(empty)
		dotnet-services	Web service development
		dotnet-signalr	SignalR development
		dotnet-webapi	WebAPI service development
		dotnet-webforms	Webform development
		dotnet-webpages	Web pages development
dotnet-whatsnew	.NET What's new	(empty)	(empty)
dual-screen	Dual-Screen	(empty)	(empty)
		microsoft-graph	Microsoft Graph
dynamics	Dynamics	(empty)	(empty)
dynamics-365	Dynamics 365	(empty)	(empty)
		microsoft-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
dynamics-365-business-central	Dynamics 365 Business Central	(empty)	(empty)
		microsoft-graph	Microsoft Graph
dynamicsax2009	z_Dynamics AX 2009	(empty)	(empty)
dynamicsax-2012	Dynamics AX 2012	(empty)	(empty)
dynamics-ax-2012	Dynamics AX 2012	(empty)	(empty)
dynamics-c5	Dynamics C5	(empty)	(empty)
dynamicscrm-2011	Dynamics CRM 2011	(empty)	(empty)
dynamicscrm-2013	Dynamics CRM 2013	(empty)	(empty)
dynamicscrm-2015	Dynamics CRM 2015	(empty)	(empty)
dynamicscrm-2016	Dynamics CRM 2016	(empty)	(empty)
dynamicscrm4	Dynamics CRM 4.0	(empty)	(empty)
dynamicsgp	Dynamics GP	(empty)	(empty)
dynamics-gp	Dynamics GP	(empty)	(empty)
dynamics-management-reporter	Dynamics	(empty)	(empty)
dynamicsnav2009	z_Dynamics NAV 2009	(empty)	(empty)
dynamicsnav2009R2	z_Dynamics NAV 2009	(empty)	(empty)
dynamicsnav2013	z_Dynamics NAV 2013	(empty)	(empty)
dynamicsnav2013R2	z_Dynamics NAV 2013	(empty)	(empty)
dynamicsnav2015	z_Dynamics NAV 2015	(empty)	(empty)
dynamicsnav2016	z_Dynamics NAV 2016	(empty)	(empty)
		microsoft-edge	Microsoft Edge
		microsoft-graph	Microsoft Graph
dynamics-nav-2016	Dynamics NAV 2016	(empty)	(empty)
dynamics-nav-2017	Dynamics NAV 2017	(empty)	(empty)
dynamics-nav-2018	Dynamics NAV 2018	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
dynamics-rmsp	Dynamics Retail Management System	(empty)	(empty)
dynamics-sl-2011	Dynamics SL 2011	(empty)	(empty)
dynamics-sl-2015	Dynamics SL 2015	(empty)	(empty)
dynamics-sl-2018	Dynamics SL 2018	(empty)	(empty)
edge	Edge	(empty)	(empty)
		microsoft-edge	Microsoft Edge
ediscovery	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
education	Education	(empty)	(empty)
		microsoft-graph	Microsoft Graph
ef-core-api	EF Core API reference	(empty)	(empty)
enterprise-search	Microsoft Enterprise Search	(empty)	(empty)
entity-framework	Entity Framework	(empty)	(empty)
		aspnet	ASP.NET
		entity-framework-6	Entity Framework 6.x
		entity-framework-core	Entity Framework Core
entity-framework-core	Entity Framework Core	(empty)	(empty)
		aspnet	ASP.NET
excel	Excel	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		data-visualizations	Data visualizations
		excel-xll	Excel XLL Software Development Kit
		fast-search	FAST search
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		scripts	Scripts
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
exchange	Exchange	(empty)	(empty)
		backup-restore	Backup/restore
		ews	Exchange web services
		management	Management
		transport-agents	Transport agents
exchange-2000	Exchange Server 2000	(empty)	(empty)
exchange-2003	Exchange Server 2003	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
exchange-2007	Exchange Server 2007	(empty)	(empty)
exchange-2010	Exchange Server 2010	(empty)	(empty)
exchange-server-2003	Exchange Server 2003	(empty)	(empty)
exchange-server-2007	Exchange Server 2007	(empty)	(empty)
exchange-server-2010	Exchange Server 2010	(empty)	(empty)
exchange-server-2013	Exchange Server 2013	(empty)	(empty)
		server-general	Server General
exchange-server-2016	Exchange Server 2016	(empty)	(empty)
exchange-server-it-pro	Exchange Server IT Pro	(empty)	(empty)
expression-studio-2	z_Expression Studio 2	(empty)	(empty)
expression-studio-3	z_Expression Studio 3	(empty)	(empty)
expression-studio-4	z_Expression Studio 4	(empty)	(empty)
extensions	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
files	OneDrive	(empty)	(empty)
		microsoft-graph	Microsoft Graph
financials	Dynamics 365 Business Central	(empty)	(empty)
		microsoft-graph	Microsoft Graph
flow	Flow	(empty)	(empty)
forms-server-2007	Forms Server 2007	(empty)	(empty)
fslogix	FSLogix	(empty)	(empty)
		application-masking	Application Masking
		java-version-control	Java Version Control
		profile-container	Profile Container
gaming	Microsoft Gaming	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
gdk	Microsoft Game Core Development Kit	(empty)	(empty)
governance	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
groove-server-2007	Groove Server 2007	(empty)	(empty)
groove-server-2010	Groove Server 2010	(empty)	(empty)
		microsoft-graph	Microsoft Graph
groups	Groups	(empty)	(empty)
		microsoft-graph	Microsoft Graph
healthvault	HealthVault	(empty)	(empty)
		windows	Windows
hololens	HoloLens	(empty)	(empty)
		windows	Windows
host-integration-server	Host Integration Server	(empty)	(empty)
hpcpack	Microsoft HPC Pack	(empty)	(empty)
hpc-server-2012-R2-and-2012	z_HPC Server 2012 R2 and 2012	(empty)	(empty)
identity-and-access	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
identity-and-access-reports	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
identity-and-sign-in	Azure AD	(empty)	(empty)
		microsoft-graph	Microsoft Graph
identity-ata	Advanced Threat Analytics	(empty)	(empty)
identity-cas	Cloud App Security	(empty)	(empty)
identity-manager-2015	Microsoft Identity Manager	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		internet-explorer	Internet Explorer
ie11	Internet Explorer 11	(empty)	(empty)
		internet-explorer	Internet Explorer
iis	Internet Information Services (IIS)	(empty)	(empty)
		iis	IIS
		iis-administration	IIS Administration API
		iis-appfx	IIS Application Frameworks
		iis-config	IIS Configuration Reference
		iis-develop	IIS Develop
		iis-extensions	IIS Extensions
		iis-hosting	IIS Web Hosting
		iis-install	IIS - Installation
		iis-manage	IIS Management
		iis-media	IIS Media Services
		iis-publish	IIS Deployment
		iis-troubleshoot	IIS Troubleshoot
infer-dotnet	Infer.NET	(empty)	(empty)
infopath	InfoPath	(empty)	(empty)
		dev-overview	Developer overview
		external-automation	External automation
		form-templates	Form templates
		microsoft-graph	Microsoft Graph
insights	Insights	(empty)	(empty)
		microsoft-graph	Microsoft Graph
intelligent-asset-mgr	Intelligent Asset Manager	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
internet-explorer	z_Internet Explorer	(empty)	(empty)
internet-explorer-10	z_IE 10	(empty)	(empty)
internet-explorer-7	z_IE 7	(empty)	(empty)
internet-explorer-8	z_IE 8	(empty)	(empty)
internet-explorer-9	z_IE 9	(empty)	(empty)
		microsoft-graph	Microsoft Graph
intune	Intune	(empty)	(empty)
		microsoft-graph	Microsoft Graph
java	Java	(empty)	(empty)
javascript	JavaScript	(empty)	(empty)
kaizala	Kaizala	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
kinect-dk	Azure Kinect	(empty)	(empty)
learning-aspnetcore	MSFT Learn: ASPNET Core	(empty)	(empty)
learning-azure	MSFT Learn: Azure	(empty)	(empty)
learning-d365-customer-service	MSFT Learn: Dynamics 365 for Customer Service	(empty)	(empty)
learning-d365-field-service	MSFT Learn: Dynamics 365 for Field Service	(empty)	(empty)
learning-flow	MSFT Learn: Flow	(empty)	(empty)
learning-powerapps	MSFT Learn: PowerApps	(empty)	(empty)
learning-power-bi	MSFT Learn: Power BI	(empty)	(empty)
legal	Legal	(empty)	(empty)
		microsoft-edge	Microsoft Edge
		uwp	UWP
		windows	Windows (general)
		xbox	Xbox
lifecycle	Microsoft Lifecycle	(empty)	(empty)
linkedin-learning	LinkedIn Learning	(empty)	(empty)
live-meeting	Live Meeting	(empty)	(empty)
		dev-overview	Developer overview
		schema	Schema

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sdk	SDK
		sdn	Software Defined Networking
lync	Lync	(empty)	(empty)
		dev-overview	Developer overview
		schema	Schema
		sdk	SDK
		sdn	Software Defined Networking
lync-2010	Lync 2010	(empty)	(empty)
lync-2013	Lync 2013	(empty)	(empty)
lync-server-2010	Lync Server 2010	(empty)	(empty)
lync-server-2013	Lync Server 2013	(empty)	(empty)
		server-general	Server General
m365-security	Microsoft 365 security	(empty)	(empty)
		m365d	Microsoft 365 Defender
		mde	Defender for Endpoint
		mdo	Defender for Office 365
		other	other
machine-learning	Machine Learning	(empty)	(empty)
mail	Outlook	(empty)	(empty)
		microsoft-graph	Microsoft Graph
mentions	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
microsoft-365	Microsoft 365	(empty)	(empty)
		microsoft-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
microsoft-365-business	Microsoft 365 Business	(empty)	(empty)
microsoft-365-enterprise	Microsoft 365 Enterprise	(empty)	(empty)
microsoft-365-lighthouse	Microsoft 365 Lighthouse	(empty)	(empty)
microsoft-365-usage-reports	Microsoft 365	(empty)	(empty)
		microsoft-graph	Microsoft Graph
microsoft-edge	Microsoft Edge	(empty)	(empty)
		chakra	Chakra
		devtools	Dev tools
		edgehtml	Edge HTML
		extensions	Extensions
		pwa	PWA
		webview	Web view
		windows-integration	Windows integration
microsoft-esp	Microsoft ESP	(empty)	(empty)
microsoft-forms	Microsoft Forms	(empty)	(empty)
microsoft-hyper-v-server-2008-R2	Microsoft Hyper-V Server 2008 R2	(empty)	(empty)
		microsoft-hyper-v-server-2012-R2-and-hyper-v-server-2012	Microsoft Hyper-V Server 2012 R2 and Hyper-V Server 2012
microsoft-hyper-v-server-2012-R2-and-hyper-v-server-2012	Microsoft Hyper-V Server 2012 R2 and Hyper-V Server 2012	(empty)	(empty)
microsoft-identity-manager	Microsoft Identity Manager	(empty)	(empty)
microsoft-identity-platform	Microsoft Identity Platform	(empty)	(empty)
		microsoft-graph	Microsoft Graph
microsoft-iot-central	Microsoft IoT Central	(empty)	(empty)
		deployr	Deploy R
		r-client	R Client

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		r-server	R Server
microsoft-r	Microsoft R	(empty)	(empty)
		deployr	Deploy R
		r-client	R Client
		r-server	R Server
microsoft-robotics	Microsoft Robotics	(empty)	(empty)
		microsoft-graph	Microsoft Graph
microsoft-teams	Microsoft Teams	(empty)	(empty)
		microsoft-graph	Microsoft Graph
mim	Microsoft Identity Manager	(empty)	(empty)
mixed-reality	MixedReality	(empty)	(empty)
		enthusiast-guide	Enthusiast Guide
		mr-dev	MR development
mlserver	Machine Learning Server	(empty)	(empty)
mse	Social Engagement	(empty)	(empty)
msix	MSIX	(empty)	(empty)
ms-stream	Microsoft Stream	(empty)	(empty)
msteams	Microsoft Teams	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
Mya	MyAnalytics	(empty)	(empty)
nimbusml	NimbusML	(empty)	(empty)
non-product-specific	Non Product Specific	(empty)	(empty)
		actionable-messages	Actionable messages
		add-ins	Add-ins
		azure-ad	Azure AD
		connectors	Connectors
		oauth-2.0	OAuth 2.0
		VBA	VBA
notes	OneNote	(empty)	(empty)
		microsoft-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
notifications	Office	(empty)	(empty)
		microsoft-graph	Microsoft Graph
nuget	NuGet	(empty)	(empty)
odata	Odata	(empty)	(empty)
		dev-overview	Developer overview
		integration	Integration
		o365-connectors	Office 365 Connectors
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		office-add-ins	Office add-ins
		office-ui-fabric	Office UI Fabric
		open-xml	Open XML
		telemetry	Telemetry
		vba	VBA
office	Office	(empty)	(empty)
		dev-overview	Developer overview
		integration	Integration
		o365-connectors	Office 365 Connectors
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		office-add-ins	Office add-ins
		office-ui-fabric	Office UI Fabric
		open-xml	Open XML
		telemetry	Telemetry

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vba	VBA
office 365	Office 365	(empty)	(empty)
		fabric	Fabric
office-2000	Office 2000	(empty)	(empty)
office-2007	Office 2007	(empty)	(empty)
office-2010	Office 2010	(empty)	(empty)
office-2013	Office 2013	(empty)	(empty)
office365	Office 365	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
office-365	Office 365	(empty)	(empty)
		microsoft-graph	Microsoft Graph
office-for-mac-2011	Office For MAC 2011	(empty)	(empty)
office-online-server	Office Online Server	(empty)	(empty)
office-online-server-powershell	Office Online Server PowerShell	(empty)	(empty)
office-perpetual-itpro	Office perpetual (IT Pro)	(empty)	(empty)
office-talk-2003	Office Talk 2003	(empty)	(empty)
office-talk-2007	Office Talk 2007	(empty)	(empty)
office-talk-2010	Office Talk 2010	(empty)	(empty)
office-vba	Office VBA	(empty)	(empty)
office-web-app-server	Office Web App Server	(empty)	(empty)
		server-general	Server General
office-xp	Office XP	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
onedrive	OneDrive	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
one-drive	OneDrive	(empty)	(empty)
onedrive-live	OneDrive Live	(empty)	(empty)
onenote	OneNote	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		dev-overview	Developer overview
		fast-search	FAST search
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		ms-graph	Microsoft Graph
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oath-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		tabs	Tabs
		vba	VBA
		webhooks	Microsoft Graph webhooks
		(empty)	(empty)
		microsoft-graph	Microsoft Graph
		(empty)	(empty)
		microsoft-graph	Microsoft Graph
		(empty)	(empty)
		microsoft-graph	Microsoft Graph
		(empty)	(empty)
openspecs-exchange	Open Specifications - Exchange	(empty)	(empty)
openspecs-ie	Open Specifications - IE	(empty)	(empty)
openspecs-office	Open Specifications - Office	(empty)	(empty)
openspecs-sql	Open Specifications - SQL	(empty)	(empty)
openspecs-windows	Open Specifications - Windows	(empty)	(empty)
outlook	Outlook	(empty)	(empty)
		actionable-messages	Actionable messages
		add-ins	Add-ins
		auxiliary	Auxiliary API

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		dev-overview	Developer overview
		fast-search	FAST search
		mapi	Messaging API
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-connectors	Office 365 Connectors
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-add-ins	Office add-ins
		office-ui-fabric	Office UI Fabric
		pia	Primary interop assembly
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		social-connector	Social connector

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		tabs	Tabs
		vba	VBA
		weather	Weather API
outlook-2000	Outlook 2000	(empty)	(empty)
outlook-2003	Outlook 2003	(empty)	(empty)
outlook-2007	Outlook 2007	(empty)	(empty)
outlook-2010	Outlook 2010	(empty)	(empty)
		partner-center-sdk	Partner Center SDK
partner-center	Partner Center developer	(empty)	(empty)
		partner-center-sdk	Partner Center SDK
people	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
people-and-workplace-intelligence	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
performance-point-server-2007	PerformancePoint Server 2007	(empty)	(empty)
personal-contacts	Outlook	(empty)	(empty)
		microsoft-graph	Microsoft Graph
places	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
planner	Planner	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		msgraph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
playfab	PlayFab	(empty)	(empty)
		drm	Digital Rights Management
		microsoft-graph	Microsoft Graph
		uwp	Universal Windows Platform
playready	Microsoft PlayReady	(empty)	(empty)
		drm	Digital Rights Management
plumbago	Plumbago	(empty)	(empty)
		uwp	Universal Windows Platform

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
power-apps	PowerApps	(empty)	(empty)
		microsoft-graph	Microsoft Graph
powerbi	PowerBI	(empty)	(empty)
		actionable-messages	Actionable messages
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
powerpoint	PowerPoint	o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
		(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		dev-overview	Developer overview
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
powerquery	Power Query	(empty)	(empty)
powershell	PowerShell Core	(empty)	(empty)
		azure-powershell	Azure PowerShell
		developer	Developing PowerShell

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dsc	Desired State Configuration
		gallery	PowerShell Gallery
		jea	Just Enough Administration
		mamccrea	PowerShell Scripting
		powershell-cmdlets	PowerShell Cmdlets
		powershell-conceptual	PowerShell Conceptual
		powershell-developer	PowerShell SDK
		powershell-dsc	PowerShell DSC
		powershell-gallery	Gallery
		powershell-jea	PowerShell JEA
		powershell-wmf	Windows Management Framework
		(empty)	(empty)
powershell-1.0	z_Windows PowerShell 1.0	(empty)	(empty)
presence	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
profile	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
project	Project	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		dev-overview	Developer overview

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		fast-search	FAST search
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
project-rome	Project Rome	(empty)	(empty)
		microsoft-graph	Microsoft Graph
project-server	Project Server	(empty)	(empty)
		vba	VBA
project-server-2007	Project Server 2007	(empty)	(empty)
project-server-2010	Project Server 2010	(empty)	(empty)
project-server-2013	Project Server 2013	(empty)	(empty)
project-server-2016	Project Server 2016	(empty)	(empty)
project-server-itpro	Project Server IT Pro	(empty)	(empty)
publisher	Publisher	(empty)	(empty)
		add-ins	Add-ins

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		tabs	Tabs
		vba	VBA
		webhooks	SharePoint webhooks
qsharp	Q#	(empty)	(empty)
quantum	Quantum Development Kit	(empty)	(empty)
		devlang-qsharp	Q#
		r-services	R Server
reporting-services	SQL Reporting Services	(empty)	(empty)
		application-integration	Application Integration

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		custom-assemblies	Custom Assemblies
		custom-report-items	Custom Report Items
		developer	Developer
		extensions	Extensions
		install-windows	Install Windows
		mobile-reports	Mobile Reports
		report-builder	Report Builder
		report-data	Report Data
		report-design	Report Design
		reporting-services	Reporting Services
		reports	Reports
		report-server	Report Server
		report-server-sharepoint	Report Server Sharepoint
		report-server-web-service	Report Server Web Service
		report-server-web-service-net-framework-exception-handling	Report Server Web Service Net Framework Exception Handling
		report-server-web-service-net-framework-soap-headers	Report Server Web Service Net Framework Soap Headers
		security	Security
		subscriptions	Subscriptions
		tools	Tools
		troubleshooting	Troubleshooting
		wmi-provider-library-reference	WMI Provider Report Server Library Reference
reporting-services-2014	SQL Reporting Services	(empty)	(empty)
		application-integration	Application Integration

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		custom-assemblies	Custom Assemblies
		custom-report-items	Custom Report Items
		developer	Developer
		extensions	Extensions
		install-windows	Install Windows
		microsoft-graph	Microsoft Graph
		mobile-reports	Mobile Reports
		report-builder	Report Builder
		report-data	Report Data
		report-design	Report Design
		reporting-services	Reporting Services
		reports	Reports
		report-server	Report Server
		report-server-sharepoint	Report Server Sharepoint
		report-server-web-service	Report Server Web Service
		report-server-web-service-net-framework-exception-handling	Report Server Web Service Net Framework Exception Handling
		report-server-web-service-net-framework-soap-headers	Report Server Web Service Net Framework Soap Headers
		security	Security
		subscriptions	Subscriptions
		tools	Tools
		troubleshooting	Troubleshooting
		wmi-provider-library-reference	WMI Provider Report Server Library Reference
reports	Reports	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		microsoft-graph	Microsoft Graph
reunion	Windows Reunion	(empty)	(empty)
		app-lifecycle	App lifecycle
		deployment	Deployment
		dwrite-core	DWrite core
		get-started	Get started
		interactive-scene-graph	Interactive Scene Graph
		mrt-core	MRT core
		power-state-notifications	Power state notifications
		push-notifications	Push notifications
		reunion-reference	Reunion API reference
		windowing	Windowing
		winui-conceptual	WinUI conceptual
		winui-reference	WinUI reference
r-server	Microsoft R	(empty)	(empty)
		r-services	R Server
sap-gateway	SAP Gateway	(empty)	(empty)
		microsoft-graph	Microsoft Graph
schema-extentions	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
search	Office	(empty)	(empty)
		microsoft-graph	Microsoft Graph
searchserver	Search Server	(empty)	(empty)
		microsoft-graph	Microsoft Graph
search-server-2008	Search Server 2008	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
search-server-2010	Search Server 2010	(empty)	(empty)
security	Security	(empty)	(empty)
		microsoft-graph	Microsoft Graph
seller-center	Seller Center	(empty)	(empty)
sharepoint	SharePoint	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		microsoft-graph	Microsoft Graph
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph	Microsoft Graph
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-add-ins	SharePoint add-ins
		sharepoint-framework	SharePoint Framework

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
sharepoint-2003	SharePoint 2003	(empty)	(empty)
sharepoint-2010	Sharepoint 2010	(empty)	(empty)
sharepoint-2016-powershell	SharePoint 2016 PowerShell	(empty)	(empty)
sharepoint-portal-server-2001	SharePoint Portal Server 2001	(empty)	(empty)
sharepoint-server-2007	SharePoint Server 2007	(empty)	(empty)
sharepoint-server-2010	SharePoint Server 2010	(empty)	(empty)
sharepoint-server-2013	SharePoint Server 2013	(empty)	(empty)
sharepoint-server-itpro	SharePoint Server IT Pro	(empty)	(empty)
sharepoint-services-3.0	SharePoint Services 3.0	(empty)	(empty)
sharepoint-team-services-1.1	SharePoint Team Services 1.1	(empty)	(empty)
shifts	Microsoft Teams	(empty)	(empty)
		microsoft-graph	Microsoft Graph
sites-and-lists	SharePoint	(empty)	(empty)
		microsoft-graph	Microsoft Graph
skype	Skype	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sdk	SDK
		sdn	Software Defined Networking
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		uri-scheme	URI scheme
		vba	VBA
		web-api	Web API
skype-2010	Skype 2010	(empty)	(empty)
skype-for-business-itpro	Skype for Business Server IT Pro	(empty)	(empty)
skype-for-business	Skype for Business	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
skype-for-business-itpro	Skype for Business Server IT Pro	(empty)	(empty)
skype-for-business-server	Skype For Business Server	(empty)	(empty)
spec-explorer	Spec Explorer	(empty)	(empty)
sql	SQL	(empty)	(empty)
		ado	ActiveX Data Objects
		analysis-services	Analysis Services (AS)
		availability-groups	availability groups
		azure-data-studio	Azure Data Studio
		azure-synapse-pathway	Azure Synapse Pathway

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		backup-restore	backup and restore
		big-data-cluster	Big Data Clusters
		clr	clr
		configuration	configuration
		connectivity	connectivity
		database-mirroring	database mirroring
		data-movement	data movement
		data-quality-services	data quality services
		data-warehouse	data warehouse
		dea	Database Experimentation Assistant
		dma	data migration assistant
		failover-cluster-instance	failover cluster instance
		filestream	filestream
		high-availability	high availability
		in-memory-oltp	In-memory OLTP
		install	install
		integration-services	SSIS
		language-extensions	language-extensions
		linux	SQL Linux
		log-shipping	log shipping
		machine-learning	Machine Learning
		machine-learning-bdc	Machine Learning on Big Data Clusters
		machine-learning-services	Machine Learning Services
		master-data-services	master data services

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
	migration-guide	Migration Guide	
	native-client	native client	
	performance	performance	
	performance-monitor	performance monitor	
	polybase	Polybase	
	profiler	profiler	
	release-landing	sql release and whats new	
	replication	replication	
	samples	samples	
	scripting	Scripting	
	search	sql search	
	security	security	
	sql-server-powershell	SQL Server PowerShell	
	ssdt	SSDT	
	ssma	SSMA	
	ssms	SSMS	
	stored-procedures	stored procedures	
	supportability	supportability	
	system-objects	system objects	
	table-view-index	tables views and indexes	
	tools-other	tools other	
	t-sql	t-sql	
	wmi	wmi	
	xevents	XEvents	
	xml	XML	

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
sql-linux	SQL Linux	(empty)	(empty)
		database-engine	Database Engine (DS)
sql-non-specified	SQL Server (no version)	(empty)	(empty)
		analysis-services	Analysis Services (AS)
		database-engine	Database Engine (DS)
		database-engine-imoltp	Database Engine - IMOTLP
		database-engine-polybase	Database Engine - Polybase
		dbe-backup-restore	Database engine - Restore
		dbe-blob	Database engine - BLOB
		dbe-bulk-import-export	Database engine - Export
		dbe-checkpoints	Database engine - Checkpoints
		dbe-cross-instance	Database engine - Instance
		dbe-databases	Database engine - Databases
		dbe-data-compression	Database engine - Compression
		dbe-data-tier-apps	Database engine - Apps
		dbe-ddl	Database engine - DDL
		dbe-dml	Database engine - DML
		dbe-high-availability	Database engine - Availability
		dbe-indexes	Database engine - Indexes
		dbe-json	Database engine - JSON
		dbe-notifications	Database engine - Notifications
		dbe-ole	Database engine - OLE
		dbe-partition	Database engine - Partition

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dbe-plan-guides	Database engine - Guides
		dbe-query-tuning	Database engine - Tuning
		dbe-search	Database engine - Search
		dbe-security	Database engine - Security
		dbe-sequence	Database engine - Sequence
		dbe-spatial	Database engine - Spatial
		dbe-statistics	Database engine - Statistics
		dbe-stretch	Database engine - Stretch
		dbe-synonyms	Database engine - Synonyms
		dbe-tables	Database engine - Tables
		dbe-transaction-log	Database engine - Log
		dbe-udf	Database engine - UDF
		dbe-views	Database engine - Views
		dbe-xml	Database engine - XML
		docset-sql-devref	SQL Developer's Reference
		drivers	SQL Server Drivers
		integration-services	Integration Services (SSIS)
		master-data-services	Master Data Services (MDS)
		replication	Replication
		reporting-services-native	Reporting Services (RS)
		sql-dma	Data Migration Assistant
		sql-ssdt	SQL Server Data Tools
		sql-ssma	SQL Server Migration Assistant
		sql-ssms	SQL Server Management Studio

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		tools-ssdt	SQL Server Data Tools
		tools-ssms	SQL Server Management Studio
sql-protocol	SQL Protocol	(empty)	(empty)
sql-server-2005	SQL Server 2005	(empty)	(empty)
sql-server-2008	SQL Server 2008	(empty)	(empty)
sql-server-2008-r2	SQL Server 2008 R2	(empty)	(empty)
sql-server-2012	SQL Server 2012	(empty)	(empty)
sql-server-2014	SQL Server 2014	(empty)	(empty)
		analysis-services	Analysis Services (AS)
		backup-restore	backup and restore
		clr	clr
		configuration	configuration
		connectivity	connectivity
		database-engine	Database Engine (DS)
		database-engine-imoltp	Database Engine - IMOTLP
		database-engine-polybase	Database Engine - Polybase
		data-movement	data movement
		data-quality-services	Data Quality Services
		dbe-backup-restore	Database engine - Restore
		dbe-blob	Database engine - BLOB
		dbe-bulk-import-export	Database engine - Export
		dbe-checkpoints	Database engine - Checkpoints
		dbe-cross-instance	Database engine - Instance
		dbe-databases	Database engine - Databases

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dbe-data-compression	Database engine - Compression
		dbe-data-tier-apps	Database engine - Apps
		dbe-ddl	Database engine - DDL
		dbe-dml	Database engine - DML
		dbe-high-availability	Database engine - Availability
		dbe-indexes	Database engine - Indexes
		dbe-json	Database engine - JSON
		dbe-notifications	Database engine - Notifications
		dbe-ole	Database engine - OLE
		dbe-partition	Database engine - Partition
		dbe-plan-guides	Database engine - Guides
		dbe-query-tuning	Database engine - Tuning
		dbe-search	Database engine - Search
		dbe-security	Database engine - Security
		dbe-sequence	Database engine - Sequence
		dbe-spatial	Database engine - Spatial
		dbe-statistics	Database engine - Statistics
		dbe-stretch	Database engine - Stretch
		dbe-synonyms	Database engine - Synonyms
		dbe-tables	Database engine - Tables
		dbe-transaction-log	Database engine - Log
		dbe-udf	Database engine - UDF
		dbe-views	Database engine - Views

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
	dbe-xml	Database engine - XML	
	docset-sql-devref	SQL Developer's Reference	
	filestream	filestream	
	high-availability	high availability	
	in-memory-oltp	In-memory OLTP	
	install	install	
	integration-services	Integration Services (SSIS)	
	master-data-services	Master Data Services (MDS)	
	native-client	native client	
	performance	performance	
	performance-monitor	Performance Monitor	
	profiler	profiler	
	release-landing	sql release and whats new	
	replication	replication	
	reporting-services	reporting-services	
	reporting-services-native	Reporting Services (RS)	
	scripting	Scripting	
	search	sql search	
	security	security	
	sql-search	search	
	sql-ssdt	SQL Server Data Tools	
	sql-ssma	SQL Server Migration Assistant	
	sql-ssms	SQL Server Management Studio	
	ssms	ssms	

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		stored-procedures	stored procedures
		supportability	supportability
		system-objects	System Objects
		table-view-index	tables views and indexes
		tools-other	Tools Other
		t-sql	t-sql
		wmi	wmi
		xevents	XEvents
		xml	xml
sql-server-2016	SQL Server 2016	(empty)	(empty)
		analysis-services	Analysis Services (AS)
		database-engine	Database Engine (DS)
		database-engine-imoltp	Database Engine - IMOTLP
		database-engine-polybase	Database Engine - Polybase
		dbe-backup-restore	Database engine - Restore
		dbe-blob	Database engine - BLOB
		dbe-bulk-import-export	Database engine - Export
		dbe-checkpoints	Database engine - Checkpoints
		dbe-cross-instance	Database engine - Instance
		dbe-databases	Database engine - Databases
		dbe-data-compression	Database engine - Compression
		dbe-data-tier-apps	Database engine - Apps
		dbe-ddl	Database engine - DDL
		dbe-dml	Database engine - DML

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dbe-high-availability	Database engine - Availability
		dbe-indexes	Database engine - Indexes
		dbe-json	Database engine - JSON
		dbe-notifications	Database engine - Notifications
		dbe-ole	Database engine - OLE
		dbe-partition	Database engine - Partition
		dbe-plan-guides	Database engine - Guides
		dbe-query-tuning	Database engine - Tuning
		dbe-search	Database engine - Search
		dbe-security	Database engine - Security
		dbe-sequence	Database engine - Sequence
		dbe-spatial	Database engine - Spatial
		dbe-statistics	Database engine - Statistics
		dbe-stretch	Database engine - Stretch
		dbe-synonyms	Database engine - Synonyms
		dbe-tables	Database engine - Tables
		dbe-transaction-log	Database engine - Log
		dbe-udf	Database engine - UDF
		dbe-views	Database engine - Views
		dbe-xml	Database engine - XML
		docset-sql-devref	SQL Developer's Reference
		integration-services	Integration Services (SSIS)
		master-data-services	Master Data Services (MDS)
		replication	Replication

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		reporting-services-native	Reporting Services (RS)
		r-services	SQL Server R Services
		sql-ssdt	SQL Server Data Tools
		sql-ssma	SQL Server Migration Assistant
		sql-ssms	SQL Server Management Studio
		xevents	Database engine - Extended Events
sql-server-2017	SQL Server 2017	(empty)	(empty)
		analysis-services	Analysis Services (AS)
		database-engine	Database Engine (DS)
		database-engine-imoltp	Database Engine - IMOTLP
		database-engine-polybase	Database Engine - Polybase
		dbe-backup-restore	Database engine - Restore
		dbe-blob	Database engine - BLOB
		dbe-bulk-import-export	Database engine - Export
		dbe-checkpoints	Database engine - Checkpoints
		dbe-cross-instance	Database engine - Instance
		dbe-databases	Database engine - Databases
		dbe-data-compression	Database engine - Compression
		dbe-data-tier-apps	Database engine - Apps
		dbe-ddl	Database engine - DDL
		dbe-dml	Database engine - DML
		dbe-high-availability	Database engine - Availability

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dbe-indexes	Database engine - Indexes
		dbe-json	Database engine - JSON
		dbe-notifications	Database engine - Notifications
		dbe-ole	Database engine - OLE
		dbe-partition	Database engine - Partition
		dbe-plan-guides	Database engine - Guides
		dbe-query-tuning	Database engine - Tuning
		dbe-search	Database engine - Search
		dbe-security	Database engine - Security
		dbe-sequence	Database engine - Sequence
		dbe-spatial	Database engine - Spatial
		dbe-statistics	Database engine - Statistics
		dbe-stretch	Database engine - Stretch
		dbe-synonyms	Database engine - Synonyms
		dbe-tables	Database engine - Tables
		dbe-transaction-log	Database engine - Log
		dbe-udf	Database engine - UDF
		dbe-views	Database engine - Views
		dbe-xml	Database engine - XML
		docset-sql-devref	SQL Developer's Reference
		integration-services	Integration Services (SSIS)
		machine-learning-services	SQL Server Machine Learning Services
		master-data-services	Master Data Services (MDS)
		replication	Replication

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		reporting-services-native	Reporting Services (RS)
		xevents	Database engine - Extended Events
sql-server-2019	SQL Server 2019	(empty)	(empty)
sql-vnext	SQL Server vNext	(empty)	(empty)
		analysis-services	Analysis Services (AS)
		database-engine	Database Engine (DS)
		database-engine-imoltp	Database Engine - IMOTLP
		database-engine-polybase	Database Engine - Polybase
		dbe-backup-restore	Database engine - Restore
		dbe-blob	Database engine - BLOB
		dbe-bulk-import-export	Database engine - Export
		dbe-checkpoints	Database engine - Checkpoints
		dbe-cross-instance	Database engine - Instance
		dbe-databases	Database engine - Databases
		dbe-data-compression	Database engine - Compression
		dbe-data-tier-apps	Database engine - Apps
		dbe-ddl	Database engine - DDL
		dbe-dml	Database engine - DML
		dbe-high-availability	Database engine - Availability
		dbe-indexes	Database engine - Indexes
		dbe-json	Database engine - JSON
		dbe-notifications	Database engine - Notifications
		dbe-ole	Database engine - OLE

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		dbe-partition	Database engine - Partition
		dbe-plan-guides	Database engine - Guides
		dbe-query-tuning	Database engine - Tuning
		dbe-search	Database engine - Search
		dbe-security	Database engine - Security
		dbe-sequence	Database engine - Sequence
		dbe-spatial	Database engine - Spatial
		dbe-statistics	Database engine - Statistics
		dbe-stretch	Database engine - Stretch
		dbe-synonyms	Database engine - Synonyms
		dbe-tables	Database engine - Tables
		dbe-transaction-log	Database engine - Log
		dbe-udf	Database engine - UDF
		dbe-views	Database engine - Views
		dbe-xml	Database engine - XML
		docset-sql-devref	SQL Developer's Reference
		integration-services	Integration Services (SSIS)
		master-data-services	Master Data Services (MDS)
		replication	Replication
		reporting-services-native	Reporting Services (RS)
		r-services	SQL Server R Services
		windows	Windows
		xevents	Database engine - Extended Events
surface	Surface	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows	Windows
surface-hub	Surface hub	(empty)	(empty)
		windows	Windows
sway	Sway	(empty)	(empty)
		uri-scheme	URI scheme
system-center	System Center	(empty)	(empty)
		app-controller	App Controller
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation	Service Manager Automation
		service-management-automation-(sma)	Service Management Automation
		service-manager	Service Manager
		service-provider-foundation	Service Provider Foundation
		virtual-machine-manager	Virtual Machine Manager
system-center-2012	System Center 2012	(empty)	(empty)
		app-controller	App Controller
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation-(sma)	Service Management Automation
		service-manager	Service Manager
		service-provider-foundation	Service Provider Foundation
		virtual-machine-manager	Virtual Machine Manager

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
system-center-2012-r2	System Center 2012	(empty)	(empty)
		app-controller	App Controller
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation-(sma)	Service Management Automation
		service-manager	Service Manager
		service-provider-foundation	Service Provider Foundation
		virtual-machine-manager	Virtual Machine Manager
system-center-2012R2	System Center 2012R2	(empty)	(empty)
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation	Service Manager Automation
		service-manager	Service Manager
		service-provider-foundation	Service Provider Foundation
		virtual-machine-manager	Virtual Machine Manager
system-center-2012-sp1	System Center 2012	(empty)	(empty)
		app-controller	App Controller
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation-(sma)	Service Management Automation
		service-manager	Service Manager

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		service-provider-foundation	Service Provider Foundation
		virtual-machine-manager	Virtual Machine Manager
system-center-threshold	System Center 2016	(empty)	(empty)
		data-protection-manager	Data Protection Manager
		operations-manager	Operations Manager
		orchestrator	Orchestrator
		service-management-automation-(sma)	Service Management Automation
		service-manager	Service Manager
		service-provider-foundation	Service Provider Foundation
		techgroup-networking	Virtual Machine Manager
		virtual-machine-manager	Virtual Machine Manager
tasks-and-plans	Planner	(empty)	(empty)
		microsoft-graph	Microsoft Graph
taxonomy	SharePoint	(empty)	(empty)
		microsoft-graph	Microsoft Graph
team-blog	Docs team blog	(empty)	(empty)
teams	Microsoft Teams	(empty)	(empty)
teamwork	Microsoft Teams	(empty)	(empty)
		microsoft-graph	Microsoft Graph
to-do-tasks	To Do	(empty)	(empty)
		microsoft-graph	Microsoft Graph
traceprocessor	.NET TraceProcessor	(empty)	(empty)
typography	Typography	(empty)	(empty)
usd1.0	Unified Service Desk 1.0	(empty)	(empty)
usd2.0	Unified Service Desk 2.0	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		apps	Apps
		app-types	App types
		audio-video-and-camera	Audio, video, and camera
		communication	Communication
		data-and-files	Data and files
		deployment	Deployment
		design	Design
		develop	Develop
		games	Games
		get-started	Get started
		microsoft-graph	Microsoft Graph
		people-and-places	People and places
		peripherals-sensors-and-power	Peripherals and sensors
		platform	Platform
		porting-apps	Porting apps
		processes-and-threading	Processes and threading
		publish	Publish
		uwp-conceptual-docs	UWP conceptual doc
		whats-new	Whats new
		winrt-reference	WinRT reference
		winui	WinUI
users	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
uwp	Universal Windows Platform (UWP)	(empty)	(empty)
		apps	Apps

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		app-types	App types
		audio-video-and-camera	Audio, video, and camera
		communication	Communication
		data-and-files	Data and files
		deployment	Deployment
		design	Design
		develop	Develop
		game-bar-sdk	Xbox Game Bar
		games	Games
		get-started	Get started
		people-and-places	People and places
		peripherals-sensors-and-power	Peripherals and sensors
		platform	Platform
		porting-apps	Porting apps
		processes-and-threading	Processes and threading
		publish	Publish
		uwp-conceptual-docs	UWP conceptual doc
		whats-new	Whats new
		winrt-reference	WinRT reference
		winui	WinUI
virtualization	Virtualization	(empty)	(empty)
visio	Visio	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		file-format	File format
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		shapesheat	Shapesheat
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
visual-basic-6	z_Visual Basic 6.0 UGH	(empty)	(empty)
visual-cpp	C++	(empty)	(empty)
		cpp-amp	AMP Library
		cpp-analysis	Analysis, Core Check
		cpp-atl	ATL Library
		cpp-azure	C++ Azure Development

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		cpp-c-language	C Language
		cpp-cli	C++/CLI Language
		cpp-concrt	Concurrency Runtime
		cpp-cx	C++/CX Language
		cpp-data	Data Access
		cpp-diagnostics	Errors and Warnings
		cpp-games	Games
		cpp-ide	IDE
		cpp-language	C++ Language
		cpp-linux	Linux Development
		cpp-masm	Microsoft Assembler
		cpp-mfc	MFC Library
		cpp-mobile	Mobile Development
		cpp-parallel	Parallel Programming
		cpp-standard-libraries	Standard Libraries
		cpp-tools	Tools
		cpp-ucrt	Universal C Runtime (UCRT)
		cpp-uwp	Universal Windows Platform
		cpp-windows	Windows Development
		cpp-winrt	C++/WinRT (Modern) Library
		vsto-add-ins	VSTO add-ins
visual-cpp-dev15	Visual C++ 2017	(empty)	(empty)
		cpp-games	Games
		cpp-ide	IDE
		cpp-language	Language

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		cpp-linux	Linux Development
		cpp-mobile	Mobile Development
		cpp-parallel	Parallel Programming
		cpp-standard-libraries	Standard Libraries
		cpp-tools	Tools
		cpp-uwp	Universal Windows Platform
		cpp-windows	Windows Development
visual-studio	Visual Studio	(empty)	(empty)
		vsto-add-ins	VSTO add-ins
		vs-tools-unity	Visual Studio Tools for Unity
visual-studio.net-2003	Visual Studio .NET 2003	(empty)	(empty)
visual-studio-2005	Visual Studio 2005	(empty)	(empty)
visual-studio-2008	z_Visual Studio 2008	(empty)	(empty)
visual-studio-2010	Visual Studio 2010	(empty)	(empty)
visual-studio-2012	Scripting 5.6	(empty)	(empty)
visual-studio-2013	Visual Studio 2013	(empty)	(empty)
visual-studio-6.0	z_Visual Studio 6.0	(empty)	(empty)
visual-studio-dev14	Visual Studio 2015	(empty)	(empty)
		cordova	Cordova
		devlang-cpp	Visual C++
		devlang-csharp	Visual C#
		devlang-fsharp	F#
		devlang-javascript	JavaScript
		devlang-python	Python
		devlang-visual-basic	Visual Basic

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		lightswitch	LightSwitch
		msbuild	MSBuild
		office-development	Office Development
		sharepoint-development	SharePoint Development
		tgt-pltfrm-aspnet	ASP 5 and Web/Cloud
		tgt-pltfrm-azure	Azure programming in VS
		tgt-pltfrm-cross-plat	Cross-Platform
		tgt-pltfrm-desktop	Desktop
		tgt-pltfrm-nodejs	Node.js
		unity	Unity
		visual-studio-sdk	Visual Studio SDK
		vs-acquisition	Installation and Getting Started
		vs-ai-tools	AI Tools
		vs-aspnet	ASP.NET and ASP.NET Core
		vs-azure	Azure Development and Tools
		vs-cordova	Cordova
		vs-cpp	Visual C++
		vs-data-tools	Data Tools
		vs-desktop	Desktop
		vs-devops-test	Continuous Delivery: Test
		vs-help-viewer	Help Viewer
		vs-ide-code-analysis	Code Analysis
		vs-ide-compile	Compile and Build
		vs-ide-debug	Debug and Diagnostics

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-ide-deployment	Deployment
		vs-ide-designers	UI Designers
		vs-ide-dotnet	.NET and .NET Core
		vs-ide-general	General IDE
		vs-ide-install	Install
		vs-ide-mobile	Mobile Development
		vs-ide-modeling	Modeling
		vs-ide-sdk	Extensibility and Integration
		vs-ide-test	Test Tools
		vs-javascript	JavaScript
		vs-lightswitch	LightSwitch
		vs-nodejs	Node.js
		vs-python	Python
		vs-rtvs	R
		vs-unity-tools	Unity Tools
		vs-uwp	UWP
		vs-workflow-designer	Workflow Designer
		vs-xml-tools	XML Tools
		xamarin-android	Android
		xamarin-ios	iOS
visual-studio-dev15	Visual Studio 2017	(empty)	(empty)
		msbuild	MSBuild
		office-development	Office Development
		sharepoint-development	SharePoint Development
		vs-acquisition	Getting Started

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-ai-tools	AI Tools
		vs-azure	Azure Development and Tools
		vs-cordova	Cordova
		vs-data-tools	Data Tools
		vs-devops-test	Continuous Delivery: Test
		vs-help-viewer	Help Viewer
		vs-ide-code-analysis	Code Analysis
		vs-ide-compile	Compile and Build
		vs-ide-debug	Debug and Diagnostics
		vs-ide-deployment	Deployment
		vs-ide-designers	UI Designers
		vs-ide-general	General IDE
		vs-ide-mobile	Mobile Development
		vs-ide-modeling	Modeling
		vs-ide-sdk	Extensibility and Integration
		vs-ide-test	Test Tools
		vs-installation	Installation
		vs-javascript	JavaScript and TypeScript
		vs-python	Python
		vs-rtvs	R
		vs-unity-tools	Unity Tools
		vs-workflow-designer	Workflow Designer
		vs-xml-tools	XML Tools
visual-studio-dev16	Visual Studio 2019	(empty)	(empty)
visual-studio-dev17	Visual Studio 2022	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
visual-studio-family	Visual Studio Family	(empty)	(empty)
		intellicode	IntelliCode
		live-share	Live Share
		visual-studio-codespaces	Visual Studio Codespaces
		vs-release-notes	VS Release Notes
		vs-subscriptions	VS Subscriptions
visual-studio-mac	Visual Studio for Mac	(empty)	(empty)
		devlang-csharp	Visual C#
		general-release-notes	General - release notes
		general-visualstudio.com	General - visualstudio.com
		tgt-pltfrm-cross-plat	Cross-Platform
		vs-ide-compile	Compile and Build
		vs-ide-debug	Debug and Diagnostics
		vs-ide-general	IDE
		vs-ide-install	Install
		vs-ide-sdk	Extensibility and Integration
visual-studio-tfs-2013	Visual Studio 2013	(empty)	(empty)
visual-studio-tfs-dev14	Visual Studio 2015	(empty)	(empty)
		devlang-cpp	Visual C +
		devlang-csharp	Visual C#
		devlang-fsharp	F#
		devlang-javascript	JavaScript
		devlang-python	Python
		devlang-visual-basic	Visual Basic
		tgt-pltfrm-aspnet	ASP 5 and Web/Cloud

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		tgt-pltfrm-azure	Azure programming in VS
		tgt-pltfrm-cross-plat	Cross-Platform
		tgt-pltfrm-desktop	Desktop
		tgt-pltfrm-nodejs	Node.js
		vs-devops-admin	Administration
		vs-devops-autotest	Testing : Automated testing
		vs-devops-build	Continuous Delivery: Build
		vs-devops-codesharing	Repository: Code sharing
		vs-devops-git	Repository: GIT
		vs-devops-integrate	Integration
		vs-devops-loadtest	Testing: Load testing
		vs-devops-manualtest	Testing : Manual testing
		vs-devops-marketplace	Marketplace
		vs-devops-onprem	TFS On-prem
		vs-devops-overview	Content.ALM Overview
		vs-devops-release	Continuous Delivery: Release management
		vs-devops-reporting	Agile-at-scale: Dashboards/reporting
		vs-devops-search	VSO/ALM Search
		vs-devops-techdebt	Technical Debt
		vs-devops-test	Continuous Delivery: Test
		vs-devops-versioncontrol	Repository: Version Control
		vs-devops-wit	Agile-at-scale : WIT
		vs-ide-compile	Compile and Build
		vs-ide-debug	Debug and Diagnostics

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-ide-deployment	Deployment
		vs-ide-general	General IDE features
		vs-ide-install	Install
		vs-ide-sdk	Visual Studio Extensibility and Integration
visual-studio-windows	Visual Studio on Windows	(empty)	(empty)
		devinit	DevInit
		msbuild	MS Build
		office-development	Office Development
		sharepoint-development	SharePoint Development
		vs-ai-tools	AI Tools
		vs-azure	Azure Development Tools
		vs-data-tools	Data Tools
		vs-help-viewer	Help Viewer
		vs-ide-code-analysis	Code Analysis
		vs-ide-compile	Compile and Build
		vs-ide-debug	Debug and Diagnostics
		vs-ide-deployment	Deployment
		vs-ide-designers	UI Designers
		vs-ide-general	General IDE
		vs-ide-mobile	Mobile Development
		vs-ide-modeling	Modeling
		vs-ide-sdk	Extensibility and Integration
		vs-ide-test	Test Tools
		vs-installation	Installation
		vs-javascript	JavaScript and TypeScript

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-python	Python
		vs-rtvs	R
		vs-unity-tools	Unity Tools
		vs-workflow-designer	Workflow Designer
		vs-xaml-tools	XAML Tools
		vs-xml-tools	XML Tools
vs-2012	Visual Studio 2012	(empty)	(empty)
vs-code	Visual Studio Code	(empty)	(empty)
vs-data	VS Data	(empty)	(empty)
		collaborate	Collaborate
		vs-devops-admin	TS Admin
		vs-devops-agile	Agile
		vs-devops-agile-wit	Agile
		vs-devops-articles	DevOps Articles
		vs-devops-build	Build
		vs-devops-extensions-api	VS Extensibility reference
		vs-devops-git	Git
		vs-devops-insights	Insights
		vs-devops-integrate	Integrate
		vs-devops-integrate-ide	VS IDE Extensions
		vs-devops-legacy	TFS 2013 and previous as well as older IDE content, but not deprecated
		vs-devops-marketplace	VS Marketplace
		vs-devops-overview	Overview or vertical-agnostic content
		vs-devops-package	Package

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-devops-release	Release Management
		vs-devops-reporting	Reporting
		vs-devops-search	Code Search
		vs-devops-setup	TS Setup
		vs-devops-techdebt	Code Insights/Technical Debt/Team Architecture
		vs-devops-test-continuous	Automated Test
		vs-devops-test-manual	Manual Test
		vs-devops-test-performance	Load and Performance Test
		vs-devops-test-tools	Test Tools
		vs-devops-tfs	TFS 2015 and 2016, current VS IDE content
		vs-devops-tfsonprem	TFS Onprem
		vs-devops-tfvc	TFVC
		vs-devops-wit	Agile
vs-devops-alm	Team Services and TFS	(empty)	(empty)
		collaborate	Collaborate
		vs-devops-admin	TS Admin
		vs-devops-agile	Agile
		vs-devops-articles	DevOps Articles
		vs-devops-build	Build
		vs-devops-extensions-api	VS Extensibility reference
		vs-devops-git	Git
		vs-devops-insights	Insights
		vs-devops-integrate	Integrate
		vs-devops-integrate-ide	VS IDE Extensions

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		vs-devops-legacy	TFS 2013 and previous as well as older IDE content, but not deprecated
		vs-devops-marketplace	VS Marketplace
		vs-devops-overview	Overview or vertical-agnostic content
		vs-devops-package	Package
		vs-devops-release	Release Management
		vs-devops-reporting	Reporting
		vs-devops-search	Code Search
		vs-devops-setup	TS Setup
		vs-devops-techdebt	Code Insights/Technical Debt/Team Architecture
		vs-devops-test-continuous	Automated Test
		vs-devops-test-manual	Manual Test
		vs-devops-test-performance	Load and Performance Test
		vs-devops-test-tools	Test Tools
		vs-devops-tfs	TFS 2015 and 2016, current VS IDE content
		vs-devops-tfsonprem	TFS Onprem
		vs-devops-tfvc	TFVC
VSscripting	z_Scripting	(empty)	(empty)
vsta-2013	z_Visual Studio Tools for Applications 2013	(empty)	(empty)
vsta-2015	z_Visual Studio Tools for Applications 2015	(empty)	(empty)
		windows	Windows
w10	Windows 10	(empty)	(empty)
		microsoft-graph	Microsoft Graph

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows	Windows
w11	Windows 11	(empty)	(empty)
web-apps-server	Office Web Apps Server	(empty)	(empty)
		hub	Hub page
		windows	Windows
whiteboard	Whiteboard	(empty)	(empty)
		microsoft-graph	Microsoft Graph
winautomation	WinAutomation	(empty)	(empty)
windows	Windows	(empty)	(empty)
		hub	Hub page
		web	Web
		windows	Windows
windows- 8.1-and-8	z_Windows 8.1 and 8.0	(empty)	(empty)
windows phone	z_Windows Phone	(empty)	(empty)
windows-10-hyperv	Hyper-V on Windows	(empty)	(empty)
windows-2000-server	Windows 2000 Server	(empty)	(empty)
windows-7	z_Windows 7	(empty)	(empty)
windows-8.1-and-8	z_Windows 8.1 and 8	(empty)	(empty)
		accessibility-insights	Accessibility Insights
windows-accessibility-tools	Windows Accessibility Tools	(empty)	(empty)
		accessibility-insights	Accessibility Insights
windows-azure-pack-for-windows-server	Windows Azure Pack	(empty)	(empty)
windows-client	Windows Client	(empty)	(empty)
		windows-client-active-directory	Active Directory

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows-client-administration-management-development	Windows Administration Management Development
		windows-client-application-compatibility	Application Compatibility
		windows-client-application-virtualization-app-v	Application Virtualization (App-V)
		windows-client-backup-and-storage	Backup and Storage
		windows-client-deployment	Deployment
		windows-client-eos	Windows EOS
		windows-client-group-policy	Group Policy
		windows-client-high-availability	High Availability
		windows-client-hyper-v	Hyper-V
		windows-client-networking	Networking
		windows-client-performance	Performance
		windows-client-printing	Printing
		windows-client-rds	RDS
		windows-client-security	Windows Security
		windows-client-shell-experience	Shell Experience
		windows-client-system-management-components	System Management Components
		windows-client-troubleshooter	Windows Troubleshooter
		windows-client-user-experience-virtualization-ue-v	User Experience Virtualization (UE-V)
		windows-client-user-profiles	User Profiles
windows-client-threshold	Windows Client Threshold	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
windows-compute-cluster	Windows Compute Cluster	(empty)	(empty)
windows-compute-cluster-pack	z_Windows Compute Cluster Pack	(empty)	(empty)
windows-compute-cluster-server-2003	Windows Compute Cluster Server 2003	(empty)	(empty)
windows-containers	Windows Containers	(empty)	(empty)
windows-desktop	Windows Desktop	(empty)	(empty)
windows-dev	Windows Developer	(empty)	(empty)
windows-dev-apps	z_Windows App Development: Windows 8 and Windows Phone Silverlight	(empty)	(empty)
windows-embedded	z_Windows Embedded	(empty)	(empty)
windows-essentials-business-server-2008	z_Windows Essentials Business Server	(empty)	(empty)
windows-essentials-server-2008	Windows Essentials Server 2008	(empty)	(empty)
windows-hardware	Windows Hardware	(empty)	(empty)
		3dprint	3dprint
		acpi	acpi
		audio	audio
		battery	battery
		biometric	biometric
		bluetooth	bluetooth
		bringup	bringup
		cfn	cfn
		dashboard	dashboard
		debugger	debugger
		devapps	devapps

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		develop	develop
		devtest	devtest
		display	display
		driversecurity	driversecurity
		gettingstarted	gettingstarted
		gnss	gnss
		gpio	gpio
		gpiobtn	gpiobtn
		hid	hid
		ieee	ieee
		ifs	ifs
		image	image
		install	install
		kernel	kernel
		mobilebroadband	mobilebroadband
		multipfunction	multipfunction
		netcx	netcx
		network	network
		nfc	nfc
		parports	parports
		partnerapps	partnerapps
		pci	pci
		pcmcia	pcmcia
		portable	portable
		pos	pos

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		powermeter	powermeter
		print	print
		samples	samples
		sd	sd
		sensors	sensors
		serports	serports
		sfu	sfu
		smartcard	smartcard
		spb	spb
		storage	storage
		stream	stream
		taef	taef
		usbcon	usbcon
		wdf	wdf
		wdk-api-reference	wdk-api-reference
		wdtf	wdtf
		whea	whea
		windows-drivers-conceptual-docs	windows-drivers
		windows-oem	Windows OEM
windows-home-server	z_Windows Home Server	(empty)	(empty)
windows-hpc-server-2008	z_Windows HPC Server 2008	(empty)	(empty)
windows-hpc-server-2008R2	z_Windows HPC Server 2008 R2	(empty)	(empty)
windows-hpc-server-2008-R2-and-2008	z_Windows HPC Server 2008 R2 and 2008	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
windows-hpc-server-2012-R2-and-2012	HPC Server 2012 R2 and 2012	(empty)	(empty)
windows-internet-explorer	z_Windows Internet Explorer	(empty)	(empty)
		iot	Internet of Things
windows-iot	Windows IoT	(empty)	(empty)
		iot	Internet of Things
windows-kinect	z_Windows Kinect	(empty)	(empty)
windows-multipoint-server	z_Windows Multipoint Server	(empty)	(empty)
windows-phone	z_Windows Phone	(empty)	(empty)
windows-powershell-1.0	z_Windows Powershell 1.0	(empty)	(empty)
windows-protocol	Windows Protocol	(empty)	(empty)
windows-rights-management-services-rms	z_Windows Rights Management Service	(empty)	(empty)
windows-script-interfaces	Windows Script Interfaces	(empty)	(empty)
windows-server	Windows Server	(empty)	(empty)
		ad-ds	AD DS
		ad-fs	AD FS
		administration	Administration
		archive	Archive
		essentials	Essentials
		failover-clustering	Failover Clustering
		getting-started	Getting Started
		hyper-v	Hyper-V
		manage	Manage
		networking	Networking
		remote	Remote access

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		remote-desktop-services	Remote Desktop Services
		security	Security
		storage	Storage
		storage-spaces	Storage Spaces
		troubleshoot	Troubleshoot
		upgrade	Upgrade
		virtualization	Virtualization
		windows-admin-center	Windows Admin Center
		windows-commands	Windows commands
		windows-containers	Windows Containers
		windows-server	Windows Server
		windows-server-active-directory	Active Directory
		windows-server-administration-management-development	Windows Administration Management Development
		windows-server-application-compatibility	Application Compatibility
		windows-server-backup-and-storage	Backup and Storage
		windows-server-containers	Containers
		windows-server-deployment	Deployment
		windows-server-eos	Windows EOS
		windows-server-group-policy	Group Policy
		windows-server-high-availability	High Availability
		windows-server-performance	Performance
		windows-server-printing	Printing

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows-server-rds	RDS
		windows-server-resources	Resources
		windows-server-sdn	SDN
		windows-server-security	Windows Security
		windows-server-security-and-malware	Security and Malware
		windows-server-shell-experience	Shell Experience
		windows-server-system-management-components	System Management Components
		windows-server-user-profiles	User Profiles
		wsus	WSUS
windows-server-2000	z_Windows Server 2000	(empty)	(empty)
windows-server-2003	z_Windows Server 2003	(empty)	(empty)
windows-server-2003-R2-and2003	z_Windows Home Server 2003 R2 and 2003	(empty)	(empty)
windows-server-2008	Windows Server 2008	(empty)	(empty)
windows-server-2008-R2	z_Windows Server 2008 R2	(empty)	(empty)
windows-server-2008-R2-and-2008	z_Windows Server 2008 R2 and 2008	(empty)	(empty)
windows-server-2012	Windows Server 2012	(empty)	(empty)
		active-directory-domain-services	Active Directory Domain Services
		active-directory-federation-services	Active Directory Federation Services
		server-general	Server General
		techgroup-compute	Compute
		techgroup-identity	Identity and Access Management

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-2012-essentials	Windows Server 2012	(empty)	(empty)
		active-directory-domain-services	Active Directory Domain Services
		active-directory-federation-services	Active Directory Federation Services
		techgroup-compute	Compute
		techgroup-identity	Identity and Access Management
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-2012-r2	Windows Server 2012	(empty)	(empty)
		active-directory-domain-services	Active Directory Domain Services
		active-directory-federation-services	Active Directory Federation Services
		hyper-v	Hyper-V
		server-general	Server General
		techgroup-compute	Compute
		techgroup-identity	Identity and Access Management
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-2012-R2-and-2012	z_Windows Server 2012 R2 and 2012	(empty)	(empty)
windows-server-2012-R2-and-2012	z_Windows Server 2012 R2 and 2012	(empty)	(empty)
windows-server-2012-r2-essentials	Windows Server 2012	(empty)	(empty)
		active-directory-domain-services	Active Directory Domain Services
		active-directory-federation-services	Active Directory Federation Services
		techgroup-compute	Compute
		techgroup-identity	Identity and Access Management
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-dev	Windows Server Developer	(empty)	(empty)
		active-directory-application-mode	Active Directory Application Mode
		active-directory-domain-services	Active Directory Domain Services
		active-directory-lightweight-directory-services	Active Directory Lightweight Directory Services
		active-directory-rights-management	Active Directory Rights Management
		active-directory-schema	Active Directory Schema
		asp.net	ASPNET
		boot-event-collector	Boot Event Collector

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
	cimwin32	CIMWIN32	
	compute-cluster-pack	Compute Cluster Pack	
	data-deduplication	Data Deduplication	
	dhcp-server	DHCP Server	
	distributed-file-system-namespace	Distributed File System Namespace	
	distributed-file-system-replication	Distributed File System Replication	
	distributed-transaction-coordinator	Distributed Transaction Coordinator	
	dns-client	DNS Client	
	dns-server	DNS Server	
	dotnet-ado	DOTNET ADO	
	dynamic-access-control	Dynamic Access Control	
	event-tracing	Event Tracing	
	failover-cluster-aware-patching	Failover Cluster Aware Patching	
	failover-cluster-hyperv	Failover Cluster Hyperv	
	failover-clustering	Failover Clustering	
	failover-cluster-management	Failover Cluster Management	
	failover-cluster-storage-qos	Failover Cluster Storage QoS	
	file-server-resource-manager	File Server Resource Manager	
	gateway-health-monitor	Gateway Health Monitor	
	group-policy	Group Policy	
	host-guardian-service	Host Guardian Service	
	intelligent-platform-management-interface	Intelligent Platform Management Interface	

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		internet-protocol-address-management	Internet Protocol Address Management
		iscsi-target	iSCSI Target
		microsoft-management-console	Microsoft Management Console
		microsoft-message-queuing	Microsoft Message Queuing
		network-file-system-(nfs)	Network File System (NFS)
		network-load-balancing	Network Load Balancing
		network-policy-and-access-services	Network Policy And Access Services
		offline-files	Offline Files
		remote-access	Remote Access
		remote-desktop-services	Remote Desktop Services
		remote-differential-compression	Remote Differential Compression
		server-message-block-(smb)	Server Message Block (SMB)
		shielded-vm-provisioning	Shielded VM Provisioning
		software-inventory-logging	Software Inventory Logging
		storage-replica	Storage Replica
		system-insights	System Insights
		user-access-logging	User Access Logging
		web-app-proxy	Web App Proxy
		windows-distributed-file-system-(dfs)	Windows Distributed File System (DFS)
		windows-event-collector	Windows Event Collector
		windows-management-instrumentation	Windows Management Instrumentation
		windows-remote-management	Windows Remote Management

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		windows-server-backup	Windows Server Backup
		work-folders	Work Folders
windows-server-essentials	z_Windows Server Essentials	(empty)	(empty)
windows-server-essentials-sbs	z_Windows Server Essentials SBS	(empty)	(empty)
windows-server-foundation	z_Windows Server Foundation	(empty)	(empty)
windows-server-storage-solutions	z_Windows Server Storage Solutions	(empty)	(empty)
windows-server-threshold	Windows Server Threshold	(empty)	(empty)
		active-directory-domain-services	Active Directory Domain Services
		active-directory-federation-services	Active Directory Federation Services
		compute-containers	Compute - Containers
		compute-hyper-v	Compute - Hyper-V
		hyper-v	Hyper-V
		identity-adds	Identity - ADDS
		identity-adfs	Identity - ADFS
		manage	Manage
		manage-group-policy	Manage - Group Policy
		manage-rsat	Manage - RSAT
		manage-server-manager	Manage - Server Manager
		manage-wsus	Manage - WSUS
		multipoint-services	Multipoint Services
		networking	Networking
		networking-bc	Networking - BC
		networking-da	Networking - DA

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		networking-dhcp	Networking - DHCP
		networking-dns	Networking - DNS
		networking-hv-switch	Networking - HVS
		networking-ipam	Networking - IPAM
		networking-nict	Networking - Nic Teaming
		networking-nlb	Networking - Load Balancing
		networking-qos	Networking - QOS
		networking-ras	Networking - RAS
		networking-sdn	Networking - SDN
		remote-desktop-services	Remote Desktop Services
		security	Security
		security-credentials-guard	Security - Credentials Guard
		security-device-guard	Security - Device Guard
		security-guarded-fabric	Security - Guarded Fabric
		security-remote-credentials-guard	Security - Remote Credentials Guard
		security-shielded-VMs	Security - Shielded VM
		security-windows-defender	Security - Windows Defender
		server-core	Server - Server Core
		server-desktop	Server - Desktop
		server-general	Server - General
		server-migration	Server - Migration
		server-nano	Server - Nano
		server-sbec	Server - SBEC
		storage	Storage

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		storage-deduplication	Storage - Deduplication
		storage-dfsn	Storage - DFSN
		storage-dfsr	Storage - DFSR
		storage-failover-clustering	Storage - Failover Clustering
		storage-file-systems	Storage - File Systems
		storage-fsrn	Storage - FSRM
		storage-health-service	Storage - Health Services
		storage-iscsi	Storage - ISCSI
		storage-nfs	Storage - NFS
		storage-qos	Storage - QOS
		storage-replica	Storage - Replica
		storage-sds	Storage - SDS
		storage-smb	Storage - SMB
		storage-spaces	Storage - Spaces
		storage-user-state-tech	Storage - User State Tech
		storage-work-folders	Storage - Work Folders
		techgroup-compute	Compute
		techgroup-identity	Identity and Access Management
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-update-services	z_Windows Server Update Service (WSUS)	(empty)	(empty)
windows-silverlight	Windows Silverlight	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
windows-subsystem-for-linux	Windows Subsystem for Linux	(empty)	(empty)
		system-utilities	System Utilities
windows-sysinternals	Windows Sysinternals	(empty)	(empty)
		system-utilities	System Utilities
windows-vista	z_Windows Vista	(empty)	(empty)
windows-xna	z_Windows XNA	(empty)	(empty)
windows-xp	z_Windows XP	(empty)	(empty)
word	Word	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-controls	Content controls
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA
workbooks-and-charts	Excel	(empty)	(empty)
		microsoft-graph	Microsoft Graph
wpa	Workplace Analytics	(empty)	(empty)
xamarin	Xamarin	(empty)	(empty)
		xamarin-android	Xamarin.Android
		xamarin-community-toolkit	Xamarin.Community Toolkit
		xamarin-crossplatform	Xamarin Cross Platform
		xamarin-essentials	Xamarin.Essentials
		xamarin-forms	Xamarin.Forms
		xamarin-ios	Xamarin.iOS
		xamarin-mac	Xamarin.Mac
		xamarin-skiasharp	SkiaSharp
xboxlive	Xbox Live	(empty)	(empty)
		(empty)	(empty)
yammer	Yammer	(empty)	(empty)
		add-ins	Add-ins
		azure-ad	Azure AD
		bf-azure-service	Azure Bot Service
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		data-visualizations	Data visualizations
		fast-search	FAST search
		module-extensions	Module extensions
		ms-bot-framework	Microsoft Bot Framework
		ms-graph-webhooks	Microsoft Graph webhooks
		o365-reporting-web-services	Office 365 Reporting web services
		o365-service-communications	Office 365 Service Communications
		oauth-2.0	OAuth 2.0
		office-ui-fabric	Office UI Fabric
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks
		tabs	Tabs
		vba	VBA

## ms.service

The `ms.service` value indicates the cloud service an article applies to; `ms.subservice` provides more granular detail about the specified service and can only be used if `ms.service` is also used. The following table shows valid pairs of `ms.service` and `ms.subservice`. An `ms.subservice` value of "(empty)" means `ms.service` can be used as standalone metadata without a child `ms.subservice`.

To request a new single `ms.service/ms.subservice` pair, fill out [the form for new ms.service/subservice values](#). Requests submitted by end-of-day Thursday will be processed on Friday and live in all systems by Monday. For bulk updates or if you have questions, email [docsmetamanager@microsoft.com](mailto:docsmetamanager@microsoft.com).

For more details, see the full taxonomy [Detail View](#).

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
3d-data-preparation	3D Data Preparation	(empty)	(empty)
active-directory	Active Directory	(empty)	(empty)
		app-mgmt	App management
		app-provisioning	Application provisioning

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		app-proxy	App proxy
		authentication	Authentication
		azuread-dev	Azure Active Directory for developers
		B2B	B2B
		B2C	B2C
		compliance	Compliance
		conditional-access	Conditional access
		develop	App development
		devices	Devices
		domain-services	Domain services
		enterprise-users	Enterprise users
		fundamentals	Fundamentals
		hybrid	Hybrid orgs
		identity-protection	Identity protection
		msi	Managed service identity
		pim	Privileged identity management
		report-monitor	Reporting and monitoring
		roles	Azure AD roles
		saas-app-tutorial	SaaS app tutorials
		seo-update-jan	SEO updates for January
		standards	Standards compliance
		user-help	Azure AD end-user content
		users-groups-roles	User groups roles
		verifiable-credentials	Verifiable Credentials

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
active-directory-b2c	Active Directory B2C	(empty)	(empty)
active-directory-ds	Active Directory Domain Services	(empty)	(empty)
ad-health-connect	Active Directory Federation Services	(empty)	(empty)
advisor	Advisor	(empty)	(empty)
aibuilder	AI Builder	(empty)	(empty)
alerts	Alerts	(empty)	(empty)
analysis-services	Azure Analysis Services	(empty)	(empty)
ansible	Ansible	(empty)	(empty)
api-management	API Management	(empty)	(empty)
application-gateway	Application Gateway	(empty)	(empty)
application-gateway-ingress-controller	Application Gateway Ingress Controller	(empty)	(empty)
applied-ai-services	Azure Applied AI Services	(empty)	(empty)
		forms-recognizer	Forms Recognizer
		immersive-reader	Immersive Reader
		metrics-advisor	Metrics Advisor
app-service	App Service	(empty)	(empty)
		app-service-environment	Environment
		app-service-govcloud	Gov Cloud
		app-service-linux	Linux
		app-service-stack	Stack
		app-service-webjobs	Webjobs
		app-service-webjobs-sdk	Webjobs SDK
app-service-api	API Apps	(empty)	(empty)
		(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
app-service-mobile	Mobile Apps	(empty)	(empty)
app-service-web	Web Apps	(empty)	(empty)
architecture-center	Architecture Center	(empty)	(empty)
		anti-pattern	Anti-patterns
		azure-guide	Azure-specific design guides
		best-practice	Best practices
		cloud-fundamentals	Cloud Fundamentals
		design-pattern	Design patterns
		enterprise-cloud-adoption	Enterprise Cloud Adoption
		example-scenario	Example Scenario
		guide	Fundamental design guides
		reference-architecture	Reference Architectures
		solution-idea	Solution Ideas
		well-architected	Microsoft Azure Well-Architected Framework
asc-for-iot	ASC for IoT	(empty)	(empty)
		asc-for-iot	ASC for IoT
as-hub	AS Hub	(empty)	(empty)
attestation	Azure Attestation	(empty)	(empty)
		attestation	Azure Attestation
automation	Automation	(empty)	(empty)
		change-inventory-management	Change and Inventory Management
		dsc	Desired State Configuration (DSC)
		process-automation	Process Automation
		shared-capabilities	Shared capabilities

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		update-management	Update Management
avere-vfxt	Avere	(empty)	(empty)
az-devops-project	Azure DevOps Project	(empty)	(empty)
azure	Azure	(empty)	(empty)
azure-advanced-threat-protection	Azure Advanced Threat Protection	(empty)	(empty)
azure-analysis-services	Azure Analysis Services	(empty)	(empty)
azure-app-configuration	Azure App Configuration	(empty)	(empty)
azure-arc	Azure Arc	(empty)	(empty)
		azure-arc-data	Azure Arc for data services
		azure-arc-kubernetes	Azure Arc for Kubernetes
		azure-arc-servers	Azure Arc for Servers
azure-asm	Azure ASM	(empty)	(empty)
azure-australia	Azure Australia Central Regions	(empty)	(empty)
azure-blockchain	Azure Blockchain	(empty)	(empty)
azure-cdn	Azure CDN	(empty)	(empty)
		azure-cdn-akamai	Akamai
		azure-cdn-microsoft	Microsoft
		azure-cdn-verizon	Verizon
azure-cli	CLI for Azure	(empty)	(empty)
azure-communication-services	Azure Communication Services	(empty)	(empty)
azure-custom-providers	Azure Custom Providers	(empty)	(empty)
azure-databricks	Data Bricks	(empty)	(empty)
		databricks-sql-analytics	Azure Databricks SQL Analytics
azure-dev-spaces	Azure Dev Spaces	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
azure-education	Azure for Education	(empty)	(empty)
		education-hub	Education Hub
azure-fluid	Fluid Framework	(empty)	(empty)
azure-functions	Azure Functions	(empty)	(empty)
		azure-functions-linux	Linux
		azure-functions-proxies	Proxies
		azure-functions-windows	Windows
		start-stop-vms	Start Stop VMs v2
azure-government	Azure Government	(empty)	(empty)
azure-government-secret	Azure Government Secret	(empty)	(empty)
azure-government-topsecret	Azure Government Top Secret	(empty)	(empty)
azure-import-export	Azure Import Export	(empty)	(empty)
azure-industry	Azure for Industry	(empty)	(empty)
		agriculture	Agriculture
		oil-and-gas	Oil & Gas
		retail	Retail
azure-java	Java for Azure	(empty)	(empty)
azure-maps	IoT Azure Maps	(empty)	(empty)
azure-migrate	Azure Migrate	(empty)	(empty)
		azure-migrate	Azure Migrate
azure-monitor	Azure Monitor	(empty)	(empty)
		agents	Agents
		alerts	Alerts
		application-insights	Application Insights
		autoscale	Autoscale

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		containers	Containers
		diagnostic-extension	Diagnostics Extension
		essentials	Essentials
		general	General
		insights	Insights
		logs	Logs
		metrics	Metrics
		monitor-common	Common / Shared Services
		virtual-machines	Virtual Machines
		visualizations	Visualizations
azure-netapp-files	Azure NetApp Files	(empty)	(empty)
azure-notebooks	Azure Notebooks	(empty)	(empty)
azure-object-anchors	Azure Object Anchors	(empty)	(empty)
azure-orbital	Azure Orbital	(empty)	(empty)
azure-percept	Azure Percept	(empty)	(empty)
azure-policy	Azure Policy	(empty)	(empty)
azure-portal	Azure Portal	(empty)	(empty)
azure-powershell	Powershell for Azure	(empty)	(empty)
azure-quantum	Azure Quantum	(empty)	(empty)
		computing	Computing
		core	Core service
		optimization	Optimization
		qdk	Quantum Development Kit
		qsharp-guide	Qsharp user guide
azure-redhat-openshift	Azure Red Hat OpenShift	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		azure-redhat-openshift	Azure Red Hat OpenShift
azure-remote-rendering	Azure Remote Rendering	(empty)	(empty)
azure-resource-manager	Resource Manager	(empty)	(empty)
		bicep	Bicep
		management	Management
		templates	Templates
azure-sentinel	Azure Sentinel	(empty)	(empty)
		azure-sentinel	Azure Sentinel
azure-spatial-anchors	Azure Spatial Anchors	(empty)	(empty)
azure-sql-edge	Azure SQL Edge	(empty)	(empty)
azure-stack	Azure Stack	(empty)	(empty)
		aks-hci	Azure Kubernetes Service on Azure Stack HCI
		azure-stack-hci	Azure Stack HCI
		azure-stack-hub	Azure Stack Hub
		azure-stack-hub-asdk	Azure Stack Hub ASDK
		azure-stack-hub-mdc	Azure Stack Hub MDC
azure-supportability	Azure Supportability	(empty)	(empty)
azure-video-analyzer	Azure Video Analyzer	(empty)	(empty)
		azure-video-analyzer-media	Azure Video Analyzer for Media
azure-vmware	Azure VMWare Solution	(empty)	(empty)
azure-vmware-cloudsimple	Azure VMware CloudSimple	(empty)	(empty)
azure-web-pubsub	Azure Web PubSub	(empty)	(empty)
backup	Backup	(empty)	(empty)
baremetal-infrastructure	BareMetal Infrastructure	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		baremetal-oracle	Oracle on BareMetal
		workloads	Workloads
bastion	Bastion	(empty)	(empty)
batch	Batch	(empty)	(empty)
batch-ai	Batch AI	(empty)	(empty)
best-practice	Best Practice	(empty)	(empty)
big-compute	Virtual Machines	(empty)	(empty)
bing-ads	Bing Ads API	(empty)	(empty)
bing-ads-ad-insight-service	Ad Insight Service	(empty)	(empty)
bing-ads-bulk-service	Bulk Service	(empty)	(empty)
bing-ads-campaign-management-service	Campaign Management Service	(empty)	(empty)
bing-ads-customer-billing-service	Customer Billing Service	(empty)	(empty)
bing-ads-customer-management-service	Customer Management Service	(empty)	(empty)
bing-ads-hotel-service	Hotel API Service	(empty)	(empty)
bing-ads-reporting-service	Reporting Service	(empty)	(empty)
bing-ads-scripts	Bing Ads Scripts	(empty)	(empty)
bing-ads-sdk	Bing Ads SDK	(empty)	(empty)
bing-ads-shopping-content	Content API Service	(empty)	(empty)
		bing-autosuggest	Bing Autosuggest
		bing-custom-autosuggest	Bing Custom Autosuggest
		bing-custom-image-search	Bing Custom Image Search
		bing-custom-search	Bing Custom Search
		bing-custom-video-search	Bing Custom Video Search
		bing-entity-search	Bing Entity Search

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		bing-image-search	Bing Image Search
		bing-local-business	Bing Local Business
		bing-news-search	Bing News Search
		bing-spell-check	Bing Spell Check
		bing-video-search	Bing Video Search
		bing-webmaster-api	Bing Webmaster API
		bing-web-search	Bing Web Search
bing-search-services	Bing Search Services	(empty)	(empty)
		bing-autosuggest	Bing Autosuggest
		bing-custom-autosuggest	Bing Custom Autosuggest
		bing-custom-image-search	Bing Custom Image Search
		bing-custom-search	Bing Custom Search
		bing-custom-video-search	Bing Custom Video Search
		bing-entity-search	Bing Entity Search
		bing-image-search	Bing Image Search
		bing-local-business	Bing Local Business
		bing-news-search	Bing News Search
		bing-spell-check	Bing Spell Check
		bing-video-search	Bing Video Search
		bing-visual-search	Bing Visual Search
		bing-web-search	Bing Web Search
bing-webmaster	Bing Webmaster	(empty)	(empty)
		bing-webmaster-api	Bing Webmaster API
biztalk-services	BizTalk Services	(empty)	(empty)
blueprints	Azure Blueprints	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
bonsai	Bonsai	(empty)	(empty)
bonsai-sa	Bonsai Solution Accelerators	(empty)	(empty)
bookings	Microsoft Bookings	(empty)	(empty)
bot-composer	Bot Framework Composer	(empty)	(empty)
bot-framework	Bot Framework	(empty)	(empty)
bot-service	Bot Service	(empty)	(empty)
business-applications	Business Applications	(empty)	(empty)
cache	Cache	(empty)	(empty)
capacity	Capacity	(empty)	(empty)
certification	Azure Certification Program	(empty)	(empty)
chef	Chef	(empty)	(empty)
china	21Vianet	(empty)	(empty)
clarity	Clarity	(empty)	(empty)
cloud-adoption-framework	Cloud Adoption Framework for Azure	(empty)	(empty)
		decision-guide	decision-guide
		general	general
		govern	govern
		innovate	innovate
		manage	manage
		migrate	migrate
		organize	organize
		plan	plan
		ready	ready
		secure	secure
		strategy	strategy

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
cloud-app-security	Cloud App Security	(empty)	(empty)
cloudpc	CloudPC	(empty)	(empty)
cloud-services	Cloud Services (classic)	(empty)	(empty)
		auto-os-updates	Guest OS Patching
		autoscale	Autoscale
		classic-to-arm-migration	Classic to ARM Migration
		deployment-files	Configuration & Package Files
		network-configuration	Network Configuration
		networking-configuration	Networking Configuration
		resource-health	Service Health
		storage-configuration	Storage Configuration
		vip-swap	Virtual IP Address Swap
cloud-services-extended-support	Cloud Services (Extended Support)	(empty)	(empty)
		autoscale	Autoscale
		classic-to-arm-migration	Classic to ARM Migration
cognitive-search	Cognitive Search	(empty)	(empty)
cognitive-services	Cognitive Services	(empty)	(empty)
		anomaly-detector	Anomaly Detector
		bing-autosuggest	Bing Autosuggest
		bing-custom-search	Bing Custom Search
		bing-entity-search	Bing Entity Search
		bing-image-search	Bing Image Search
		bing-local-business	Bing Local Business
		bing-news-search	Bing News Search
		bing-speech	Bing Speech

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		bing-spell-check	Bing Spell Check
		bing-video-search	Bing Video Search
		bing-visual-search	Bing Visual Search
		bing-web-search	Bing Web Search
		computer-vision	Computer Vision
		content-moderator	Content Moderator
		custom-vision	Custom Vision
		face-api	Face API
		ink-recognizer	Ink Recognizer
		language-understanding	Language Understanding (LUIS)
		personalizer	Personalizer
		qna-maker	QnA Maker
		speech-service	Speech Service
		text-analytics	Text Analytics
		translator-text	Translator Text
		video-indexer	Video Indexer
common-data-model	Common Data Model	(empty)	(empty)
commondataservice	Common Data Service	(empty)	(empty)
confidential-ledger	Confidential Ledger	(empty)	(empty)
connectors	BAPI Connectors	(empty)	(empty)
consumer	Consumer	(empty)	(empty)
		consumer	consumer
container-instances	Container Instances	(empty)	(empty)
		confidential-computing	Confidential Computing
container-registry	Container Registry	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
container-service	Azure Container Service	(empty)	(empty)
		confidential-computing	Azure Confidential Compute
cortana-analytics	Cortana Analytics	(empty)	(empty)
cosmos-db	Azure Cosmos DB	(empty)	(empty)
		cosmosdb-cassandra	Cosmos DB Cassandra
		cosmosdb-graph	Cosmos DB Graph
		cosmosdb-mongo	Cosmos DB Mongo
		cosmosdb-sql	Cosmos DB SQL
		cosmosdb-table	Cosmos DB Table
cost-management-billing	Cost Management + Billing	(empty)	(empty)
		billing	Billing
		cloudyn	Cloudyn
		common	Common
		cost-management	Cost Management
		enterprise	Enterprise Agreement
		reservations	Reservations
crm-online	Dynamics 365 (CRM)	(empty)	(empty)
csp	Cloud Solution Provider	(empty)	(empty)
customer-insights	Customer Insights	(empty)	(empty)
		audience-insights	audience insights capability
		engagement-insights	engagement insights capability
cyclecloud	CycleCloud	(empty)	(empty)
databox	Azure Data Box	(empty)	(empty)
		disk	Data Box Disk

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		edge	Data Box Edge
		edge-fpga	Azure Stack Edge with FPGA
		edge-gpu	Azure Stack Edge with GPU
		edge-r	Azure Stack Edge R-series
		gateway	Data Box Gateway
		heavy	Data Box Heavy
		pod	Data Box
data-catalog	Data Catalog	(empty)	(empty)
		data-catalog-gen1	Generation 1
		data-catalog-gen2	Generation 2
data-explorer	Azure Data Explorer	(empty)	(empty)
data-factory	Data Factory	(empty)	(empty)
		connectivity	Connectivity
		design	Design
		manage	Manage
		scripts	Scripts
dataflows	DataFlows	(empty)	(empty)
data-lake-analytics	Data Lake Analytics	(empty)	(empty)
data-lake-store	Data Lake Store	(empty)	(empty)
data-science-vm	Data Science VM	(empty)	(empty)
data-share	Azure Data Share	(empty)	(empty)
data-transfers	Data Transfers	(empty)	(empty)
ddos-protection	DDoS Protection	(empty)	(empty)
dedicated-hsm	Dedicated HSM	(empty)	(empty)
defender-for-iot	Defender for IoT	(empty)	(empty)